



**FACULTAD DE INFORMÁTICA Y CIENCIAS APLICADAS
TÉCNICO EN INGENIERÍA DE SOFTWARE**



ELABORACIÓN DE MATERIAL DIDÁCTICO DE APOYO AL DESARROLLO DE
SISTEMAS INFORMÁTICOS UTILIZANDO EL LENGUAJE DE PROGRAMACIÓN
JAVA Y LA BASE DE DATOS DE ORACLE 11G PARA EL DESARROLLO DE
COMPETENCIAS EN LA CÁTEDRA DE DESARROLLO DE SISTEMAS
INFORMATICOS.

TRABAJO DE GRADUACIÓN PRESENTADO POR:

JOSE ROBERTO RAYMUNDO TEOS

RICARDO JAVIER ELIAS AGUIRRE

PARA OPTAR AL GRADO DE:

TÉCNICO EN INGENIERÍA DE SOFTWARE

SEPTIEMBRE, 2012

SAN SALVADOR, EL SALVADOR, CENTROAMÉRICA

PAGINA DE AUTORIDADES

LIC. JOSE MAURICIO LOUCEL
RECTOR

ING. NELSON ZÁRATE SÁNCHEZ
VICERRECTOR GENERAL

ING. FRANCISCO ARMANDO ZEPEDA
DECANO

JURADO EXAMINADOR

LIC. FABRICIO QUINTANILLA
PRESIDENTE

ING. EDWING ALBERTO CALLEJAS
PRIMER VOCAL

LIC. JORGE MACHADO
SEGUNDO VOCAL

SEPTIEMBRE, 2012

SAN SALVADOR, EL SALVADOR, CENTROAMÉRICA

AGRADECIMIENTOS

Doy gracias a Dios infinitamente por haberme dado vida y oportunidad de llevar a feliz término mi carrera.

A mis Padres

Por sus buenos deseos para salir adelante en mis propósitos académicos apoyándome incondicionalmente y por los cuidados brindados cuando más los necesitaba.

A mi amada esposa

Agradezco toda su paciencia, comprensión y sacrificio que ha hecho por mí todo este tiempo de desvelos, quien ha sido una pieza clave en mi desarrollo profesional por brindarme sus conocimientos y apoyarme siempre, dándome ánimos en todo momento, es para ti este triunfo mi amor, Mil gracias.

A mis maestros y asesor

En principal mi maestro el Ing. Callejas, gracias por brindarme parte de sus conocimientos técnicos, bases, orientación para buscar información y herramientas para aprender a ejecutar las operaciones que aseguran el éxito en mi carrera y en mi vida profesional.

José Roberto Raymundo Teos

INDICE

INTRODUCCION	i
CAPITULO I	
SITUACION ACTUAL	
1.1 Situación Problemática	1
1.2 Justificación	2
1.3 Objetivos	3
1.3.1 Objetivo General	3
1.3.2 Objetivos Específicos	4
1.4 Alcances	4
1.4.1 Alcance Geográfico	5
1.4.2 Alcance Temporal	5
1.4.3 Alcance Organizacional	5
1.4.4 Alcance Funcional	5
1.5 Estudio de Factibilidades	6
1.5.1 Factibilidad Técnica	6
1.5.2 Factibilidad Económica	7
1.5.3 Factibilidad Operativa	7

1.5.4 Factibilidad Legal	8
1.6 Carta de Aceptación	9
 CAPITULO II	
DOCUMENTACION TECNICA	
2.1 Recopilación de Requisitos para uso correcto de Oracle 11g.....	10
2.1.1 Requerimientos de Hardware de Oracle.....	10
2.1.2 Requerimientos de Software de Oracle	10
2.1.3 Generalidades de Oracle	11
2.1.4 Requisitos de Instalación de Oracle	12
2.1.5 Proceso de Instalación y Configuración de Oracle	13
2.1.5.1 Guía de Instalación de Oracle	13
2.2 Recopilación de Requisitos Técnicos de Java	22
2.2.1 Requerimientos de Hardware de Java.....	22
2.2.2 Requerimientos de Software de Java	23
2.2.3 Generalidades de Java	25
2.2.4 Proceso de Instalación y Configuración de Java	25
2.2.4.1 Instalación del Jdk v.7.0 de Java, previa del NetBeans	25
2.3 Recopilación de Requisitos de NetBeans	31
2.3.1 Requerimientos de Hardware de NetBeans.....	31

2.3.2 Generalidades de NetBeans.....	32
2.3.3 Requisitos de Instalación de NetBeans	32
2.3.4 Instalación y Configuración de NetBeans	33
2.3.4.1 Instalación de NetBeans.....	33
2.4 Recopilación de Requisitos de Moodle	37
2.4.1 Requerimientos de Hardware de Moodle	37
2.4.2 Requerimientos de Software de Moodle.....	37
2.4.3 Generalidades de Moodle.....	38
2.4.4 Requisitos de Instalación de Moodle	38
2.5 Cronograma de actividades	40
2.6 Presupuesto.....	40

CAPITULO III.

DISEÑO DE LA SOLUCION

3.1 Diseño de la solución.....	41
3.2 Diseño del Material Didáctico	41
3.3 Instalación y Configuración	43
3.3.1 Configuración de Oracle	43
3.4 Diseño de Ejemplos Prácticos de Oracle.....	61
3.5 Diseño de Ejercicios Propuestos en Java.....	61

Conclusión	62
Recomendación	63
Glosario	63
Bibliografía.....	66
Anexos.....	69

INTRODUCCIÓN

El presente documento describe el proyecto titulado elaboración del material didáctico, de apoyo al desarrollo de sistemas informáticos, utilizando el lenguaje de programación Java y la base de datos de Oracle 11g, para la ejecución de competencias en la cátedra de desarrollo de sistemas informáticos; como parte del trabajo de graduación del Técnico en Ingeniería de Software en el cual; se presentan las siguientes partes.

En la primera parte del documento se muestra una situación del problema que actualmente poseen los estudiantes de la cátedra de desarrollo de sistemas, la cual, es la falta de apoyo en la realización de ejercicios y búsqueda de información en sus diferentes actividades que se le presentan a diario en su jornada estudiantil.

Posteriormente se define la justificación, donde se muestran los principales beneficios a obtener, con la elaboración del material didáctico de apoyo al desarrollo de sistemas informáticos, utilizando el lenguaje de programación Java y la base de datos de Oracle 11g, para ejecutar las competencias en la cátedra de desarrollo de sistemas.

CAPITULO I. SITUACION ACTUAL

1.1 Situación Problemática

La Universidad Tecnológica de El Salvador (UTEC), a través de la Escuela de Informática, cuenta con mucho material de apoyo didáctico para sus docentes y estudiantes, sobre el lenguaje de programación Java y la base de datos de Oracle 11g. La riqueza de información en internet y la ausencia de material directo para el desarrollo de sistemas informáticos, hace que la tarea de desarrollar sistemas de información al estudiante sea ardua y complicada por no existir una forma de buscar información relevante para el desarrollo de sistemas de información; además, la preparación del docente debe estar respaldada por un material de apoyo para poderse transmitir a sus estudiantes para que estos puedan desarrollar sistemas de información en estas tecnologías.

Durante el desarrollo de prácticas de la asignatura surgen dudas y no existe material de apoyo para poder consultar cuando el docente no está presente al momento de estar desarrollando un sistema de información; ya que estos se realizan fuera del tiempo de clase; debido a que la hora de clase es aprovechado para poder impartir nuevas temáticas, pero para desarrollar sistemas de información, afuera de la clase necesitamos soportes, guías del docente o de un material específico con el cual por el momento no se cuenta.

Muchas veces no se alcanza la comprensión y realización de ejercicios con las tecnologías, como Java y Oracle11g que hacen referencias en el material tanto en clase como en las tareas ex-aulas.

La información que se tiene de estos temas es mucha; pero el problema con el que se cuenta es que esta es muy extensa y no se tiene una referencia puntual de los temas, que se están viendo en la cátedra de desarrollo de sistemas.

1.2 Justificación

Con el desarrollo de un material de apoyo basado en tecnología de Java y la base de datos Oracle11g, para la ejecución y desarrollo de competencias de la cátedra de desarrollo de sistemas en la Escuela de Informática de la UTEC, se espera que su contenido supla las necesidades de conceptos, configuraciones, conexiones, bases de datos, e-learning, buscadores de información y la creación de aulas virtuales; facilitando material a sus docentes y estudiantes.

Los estudiantes ampliarán sus conocimientos; a través de prácticas e investigación en el aula virtual; ya que existe un gran número de aplicaciones y sitios Web que no funcionan a menos que Java esté instalado; Java es rápido, seguro, fiable y la instalación de la herramienta servidor Oracle Designer esta se conecta a la base de datos y Oracle Developer para crear formularios en local, que son las básicas en programación sobre Oracle, en ese sentido, el material propuesto vendrá a despejar muchas dudas sobre estos temas y

además a exponer su contenido a los estudiantes y docentes para profundizar en estos temas.

Con la creación de este material didáctico se proporcionara una ayuda tanto al estudiante como al docente en los temas de Oracle 11g y Java con una información puntual de estos en su instalación y desarrollo de aplicaciones en este software.

Con la elaboración del material didáctico, se obtendrá la creación y comprensión de una nueva herramienta de estudio, que facilitara al docente y al estudiante; una guía de apoyo al momento de investigar la información, que este material proporcionara en los temas de Java y Oracle. Teniendo en cuenta que actualmente no existen libros o material de apoyo para el estudiante como para el docente, con la recopilación de información, se espera llegar a cubrir esa necesidad en una asignatura y así tratar de mitigar la ausencia de material de apoyo en donde se integren estas dos tecnologías en un solo documento.

1.3 Objetivos

1.3.1 Objetivo General

Elaboración de material didáctico de apoyo al desarrollo de sistemas informáticos utilizando el lenguaje de programación Java y la base de datos de Oracle 11g, para el desarrollo de competencias en la cátedra de desarrollo de sistemas.

1.3.2 Objetivos Específicos

- Realizar una investigación de los contenidos temáticos del lenguaje Java, la Base de Datos Oracle 11g y la configuración de la herramienta de estudio Moodle.
- Elaborar un aula virtual implementando Moodle, para presentar los contenidos del material propuesto.
- Elaborar los contenidos a desarrollar en el material didáctico, implementándole el aula virtual.

1.4 Alcances

Crear un material de apoyo didáctico para la cátedra de desarrollo de sistemas informáticos, utilizando el lenguaje de programación Java y la base de datos Oracle 11g, para el desarrollo de competencias en el desarrollo de sistemas informáticos, y con la implementación del software Moodle; se espera cubrir las deficiencias del estudiante; superando sus expectativas sobre la asignatura, convirtiendo esto en un material de apoyo en sus tareas y actividades.

El material contará con la descripción de ejercicios prácticos, que hagan visibles las formas de interactuar con los ejercicios, para la ejecución de un ejemplo para cada usuario.

El material de apoyo que se desarrollara, tendrá la información necesaria para facilitar la comprensión del docente y el estudiante; a fin de que estos puedan

tener una opción más, al momento de consultar sobre los temas de Oracle y Java.

1.4.1 Alcance Geográfico

La elaboración del material se desarrollo en la UTEC, en la Escuela de Informática.

1.4.2 Alcance Temporal

La elaboración del material de apoyo al desarrollo de competencias en la cátedra de desarrollo de sistemas, comprende durante el proceso de graduación, desde febrero a junio del 2012.

1.4.3 Alcance Organizacional

El material se desarrollo bajo la responsabilidad de los estudiantes del equipo de trabajo de graduación que a continuación se detalla:

- José Roberto Raymundo Teos
- Ricardo Javier Elías Aguirre
 - Asesor Técnico: Ing. Jorge Edwin Machado Leiva
 - Asesor Metodológico: Ing. Edwin Alberto Callejas.

1.4.4 Alcance Funcional

El material didáctico contempla los siguiente puntos tales como:

1. Creación del material e Instalación del Software.
2. Instalación y configuración de herramientas.
3. Implementación de la Base de Datos.
4. Creación de aplicaciones web utilizando el lenguaje de programación Java y la Base de Datos 11G.
5. Aprender a acceder a datos mediante el uso de la sintaxis en Oracle.

1.5 Estudio de Factibilidades

1.5.1 Factibilidad Técnica

Consiste en la creación de un Material que sirva de apoyo y guía de consulta para docentes y estudiantes de la Escuela de Informática de la UTEC, para lo cual se recolecto información sobre los componentes técnicos y la posibilidad de hacer uso de los mismos en el desarrollo e implementación del proyecto asignado.

Se cuenta con equipo y las características requeridas para las instalaciones de software, tal como se muestra en la figura 1.5.1.1 y se tiene el software para el desarrollo del material.

Atributo	Descripción
Memoria RAM	2GB de Memoria RAM
Disco duro	80 Gigabytes
Microprocesador	Procesador 2 GHz mínimo
Tarjeta de red	10 Megabytes
Tarjeta de video	Tarjeta de Vídeo 256 colores
Sistema operativo	Windows XP de 32 bits
IDE	Java Net bean 7.1
Sistema gestor de Base de Datos	Oracle 11G

Figura 1.5.1.1

Características de la computadora para el desarrollo de aplicaciones

1.5.2 Factibilidad Económica

Se realizo el análisis que será más beneficioso la utilización de los software; ya que estos son de licencia libre, para la creación del material, para la UTEC no representara ningún costo.

1.5.3 Factibilidad Operativa

Se realizo entrevista con el jefe de catedráticos de la Escuela de Informática de la UTEC, quien menciona que hace falta, un material que sirva de apoyo a los docentes y estudiantes; por lo cual la creación del presente material, aportara los conocimientos necesarios sobre la base de datos de Oracle y el lenguaje de programación Java.

Se consolidara información de materiales didácticos y se crearan pantallas con descripción de los pasos, para que se lleve una guía de como ejecutar los procesos en programación.

1.5.4 Factibilidad Legal

Es factible, puesto que no se violaran derechos de autor, no se necesitara ninguna licencia, también se cuentan con los permisos de parte de la UTEC y la Escuela de Informática.

1.6 Carta de Aceptación

San Salvador, 19 de junio de 2012

Ingeniero

JORGE ARMANDO APARICIO LEMUS

Director de Escuela de Informática

Facultad de Informática y Ciencias Aplicadas

Presente

De la manera más atenta nos dirigimos a usted deseándole éxitos en la dirección de la Escuela de informática de nuestra Universidad.

El motivo de la presente, es para solicitar su autorización para realizar nuestro trabajo de graduación de la carrera de Técnico en Ingeniería de Software titulado “Elaboración de material de apoyo al desarrollo de sistemas informáticos utilizando el lenguaje de programación Java y la base de datos Oracle 11g para el desarrollo de competencias de la cátedra de desarrollo de sistemas”, durante el proceso de grado del ciclo 01-2012.

Esperamos contar con su apoyo a dicho proyecto; ya que consideramos que será de un gran beneficio el producto a obtener.

En espera de una respuesta positiva a nuestra solicitud, nos suscribimos atentamente,

José Roberto Raymundo Teos	27-1367-2002
----------------------------	--------------

Ricardo Javier Elias Aguirre	27-1535-2008
------------------------------	--------------

CAPITULO II. DOCUMENTACIÓN TÉCNICA

2.1 Recopilación de requisitos para uso correcto de Oracle 11g.

2.1.1 Requerimientos de Hardware de Oracle

Sistema operativo Windows Xp

- Requisito Mínimo Memoria Física (RAM), 1Gb
- Memoria Virtual, El doble de tamaño que la RAM (2Gb)
- Espacio en disco, Instalación Básica: 4.55Gb, Instalación Avanzada: 4.92Gb
- Adaptación de Vídeo, 256 colores
- Procesador, 550Mhz

2.1.2 Requerimientos de Software de Oracle

- Arquitectura de procesador, Intel (x86), AMD64, o Intel EM64T
- Sistema Operativo, Windows 2000 Service Pack 1 o superior, Windows Server 2003 o Superior, Windows XP Professional

Software de red para la base de datos Oracle 11g.

Cada nodo del cluster requiere un protocolo de interconexión para soportar cache fusión y Oracle Clusterware polling.

Es requerimiento que el protocolo elegido, esté certificado para la plataforma Oracle a implementar.

Paquetes requeridos para Oracle 11g

Los paquetes requeridos dependen de la instalación que se pretenda hacer.

Usuarios y grupos requeridos para la instalación de Oracle 11g

En la siguiente tabla se muestran los tipos de usuarios de Oracle:

Nombre	Tipo	Rol
oinstall	Grupo	Grupo de inventario de Oracle
dba	Grupo	Grupo de OSDBA
oracle	Usuario	El propietario del software Oracle
nobody	Usuario	Usuario sin privilegios en el sistema

El usuario Oracle deberá pertenecer al grupo oinstall (grupo primario) y también al grupo dba (grupo secundario).

2.1.3 Generalidades de Oracle

QUÉ ES ORACLE?:

Es una Base de Datos, los cuales están almacenados y accesibles según el formato de tablas relacionales. Almacenada físicamente en ficheros y la correspondencia entre los ficheros, las tablas es posible por las estructuras internas de la BD, que permiten que diferentes tipos de datos, estén

almacenados físicamente separados. Esta división lógica se hace por los espacios de tablas, tablespaces.

ORACLE SE UTILIZA:

como la base de datos más completos, aplicaciones comerciales, herramientas de desarrollo de aplicaciones, soporte de decisiones, soporte de transacciones, estabilidad, escalabilidad y Soporte multiplataforma.

2.1.4 Requisitos de Instalación de Oracle

LA INSTALACIÓN DE ORACLE 11G: es un proceso que requiere de dos fases en la que se ejecuta el Oracle Universal Installer (OUI).

- En la primera etapa se instala Oracle Clusterware Release 2 (10.2.0).
- En la segunda etapa se instala el software de base de datos para Oracle 11g.
- Completar las tareas de preinstalación
 - Requerimientos de hardware
 - Requerimientos de software
- Configuración del entorno (parámetros del kernel, etc)
- Instalar Oracle Clusterware
- Instalar ASM
- Instalar el software de base de datos Oracle 11g
- Instalar agentes de Enterprise Manager en cada nodo del cluster

- Crear la base de datos en cluster
- Completar tareas de postinstalación

2.1.5 Proceso de Instalación y configuración de Oracle

2.1.5.1 Guía de Instalación de Oracle



Figura: 2.1.5.1.1

Creación de usuario de sistema operativo

La figura 2.1.5.1.1, muestra el proceso de creación de un usuario del sistema operativo, de tipo administrador; para lo cual se selecciona en menú inicio, la opción, panel de control.

Esta cuenta, servirá para instalar el software de la base de datos; por lo que nos aparece la siguiente pantalla descrita, en la figura 2.1.5.1.2.

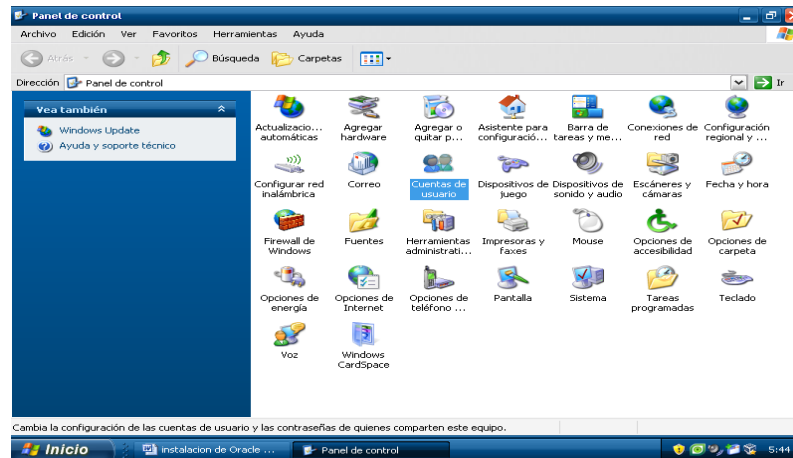


Figura 2.1.5.1.2

Creación de cuenta de usuario de sistema operativo

La figura 2.1.5.1.2, nos permite en el panel de control, crear una nueva cuenta de usuario de tipo administrador, tal como se ilustra en la figura 2.1.5.1.4, la cual le asignaremos el nombre de Oracle

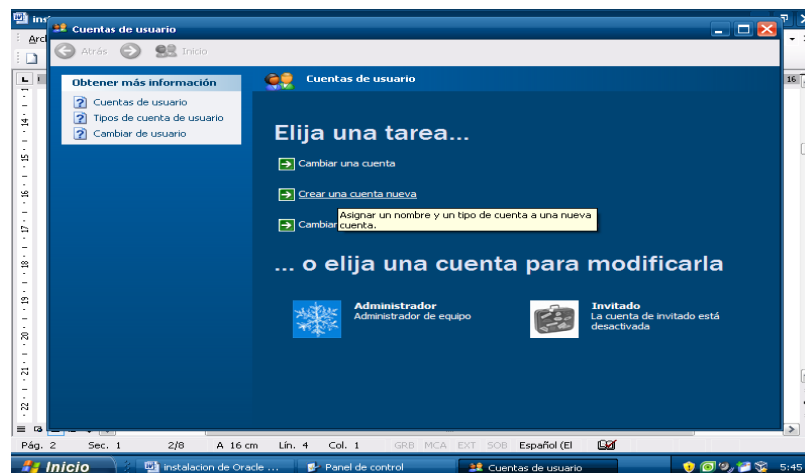


Figura 2.1.5.1.3

Es para acceder al menú de cuentas de usuario

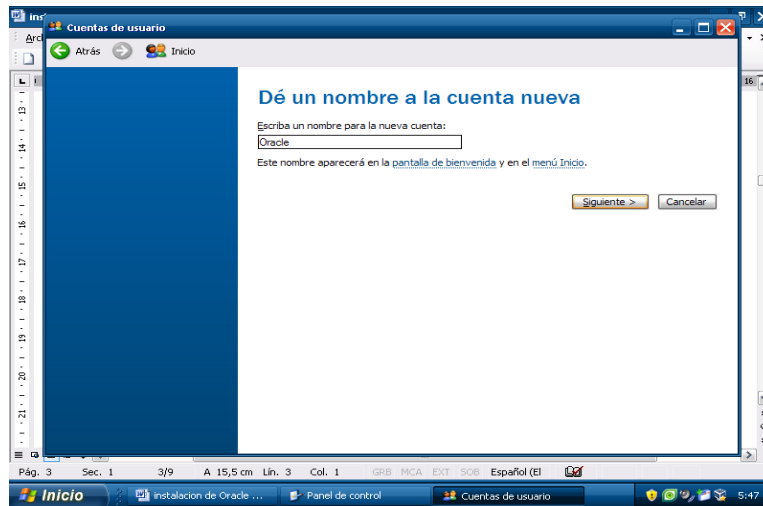


Figura 2.1.5.1.4

Creación de la cuenta de usuario y pulsar siguiente, para continuar con la creación del nuevo usuario.

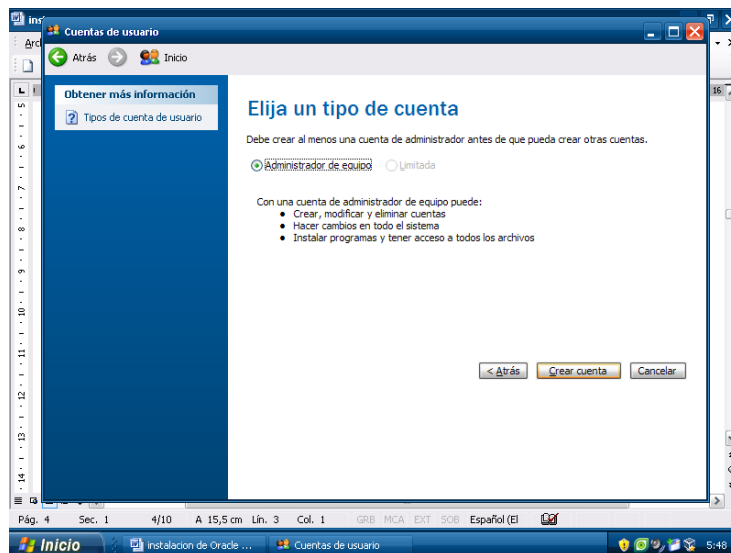


Figura 2.1.5.1.5

Hacer clic, en la opción crear cuenta.

Inicio de la instalación de Oracle 11g.

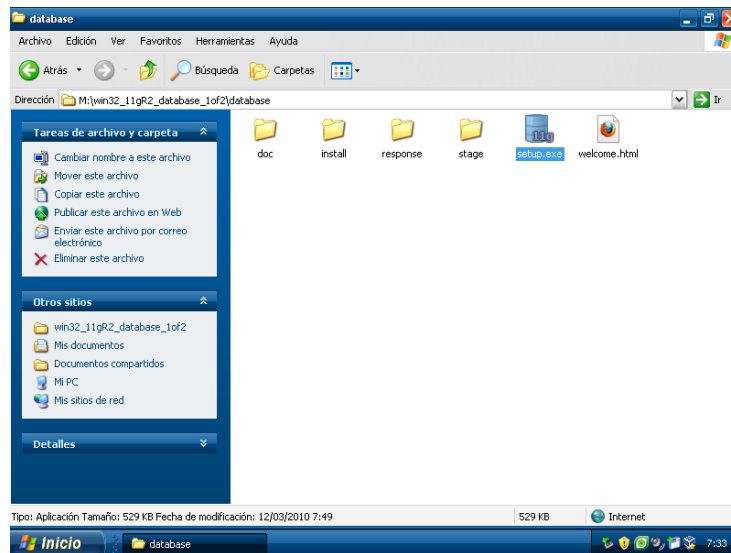


Figura 2.1.5.1.6

Ejecutar, el icono del instalador de Oracle

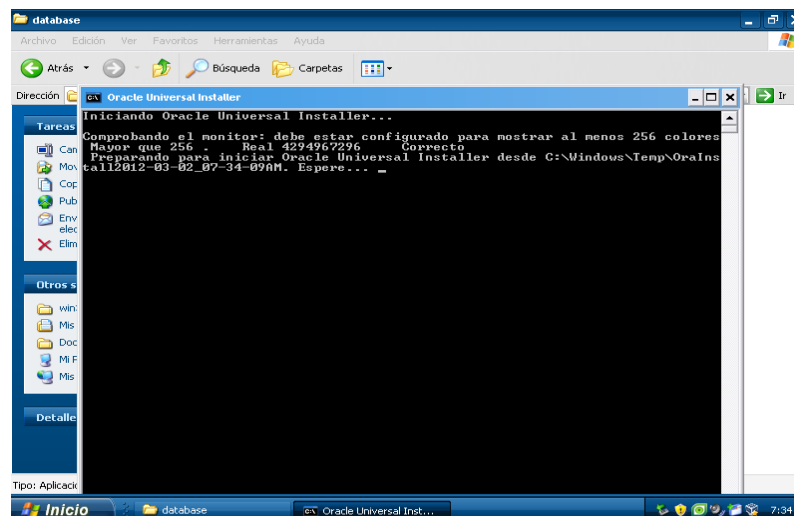


Figura 2.1.5.1.7

Deberá aparecer la siguiente pantalla de inicialización de instalación propio de Oracle



Figura 2.1.5.1.8

Proceso de Inicio.



Figura 2.1.5.1.9

Inicio del asiste de instalación.

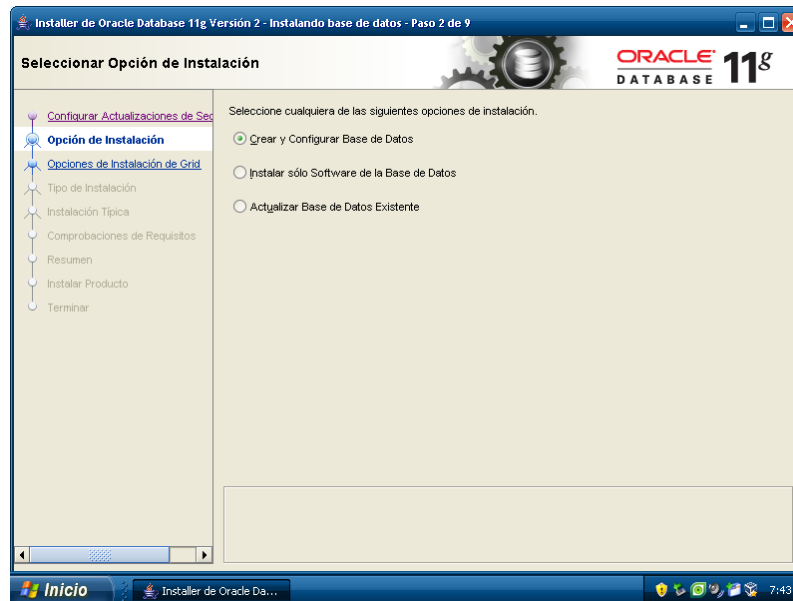


Figura 2.1.5.1.10

Iniciara el asistente para la configuración de instalación de Oracle, el asistente será la guía para la instalación

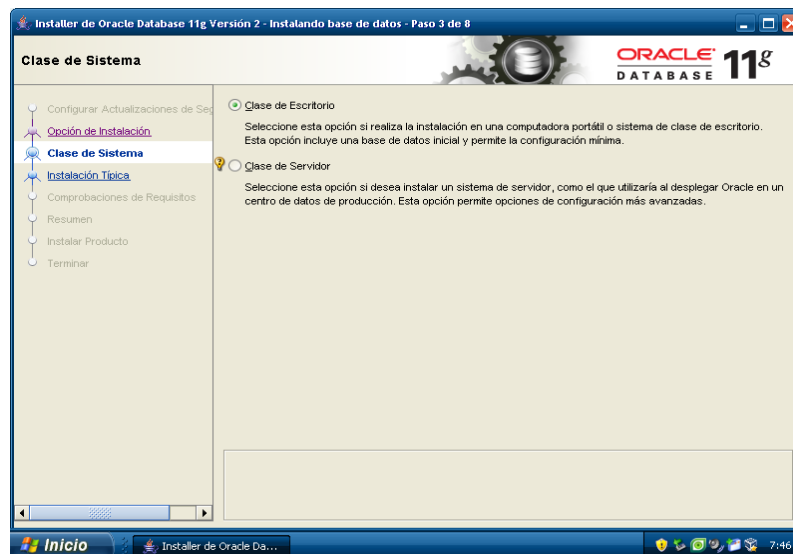


Figura 2.1.5.1.11

En esta opción elegir, la clase del sistema a instalar.

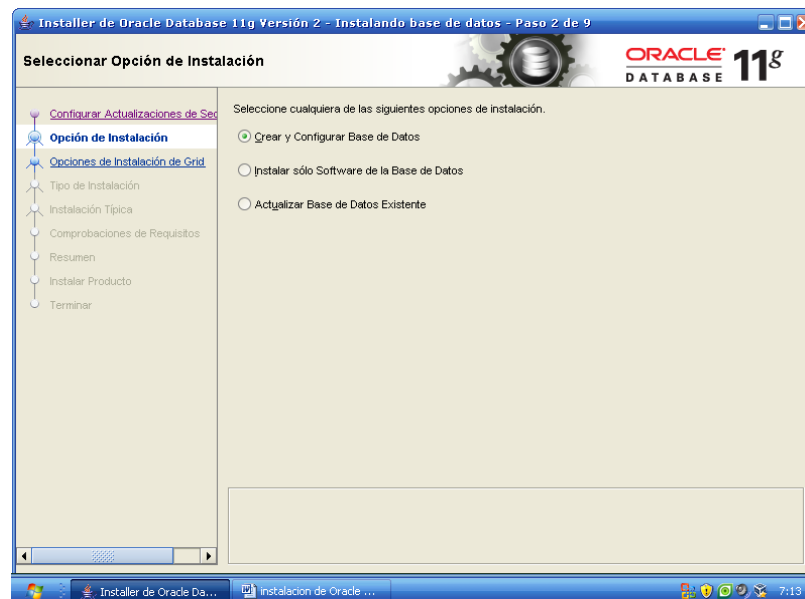


Figura 2.1.5.1.12

Seleccionar la opción, del tipo de instalación, de la base de datos

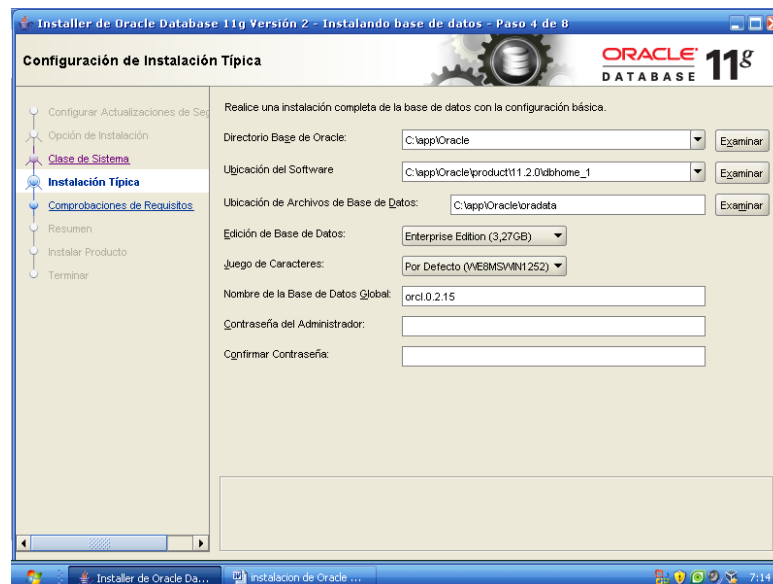


Figura 2.1.5.1.13

Se administra la contraseña

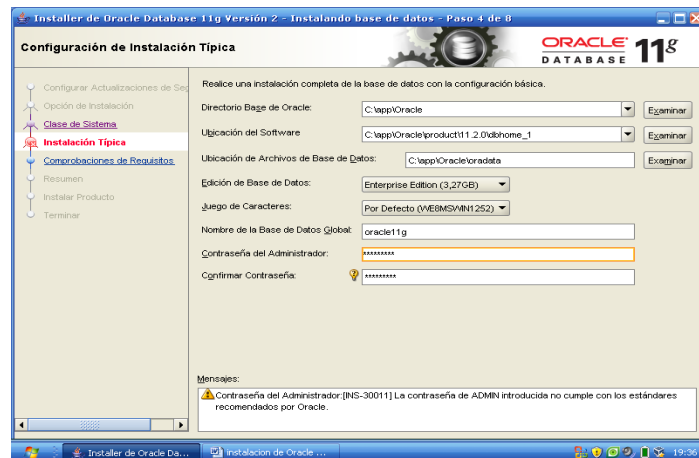


Figura 2.1.5.1.14

Se Iniciar la configuración, de la base de datos.

Al asignar la contraseña, se debe tomar en cuenta que Oracle, recomienda ocho caracteres como mínimo, usar letras mayúsculas y números.

Si alguna de estas características, no se cumplen, aparecerá la siguiente pantalla, en la cual se debe elegir, si continuar con la instalación o cambiar la contraseña.

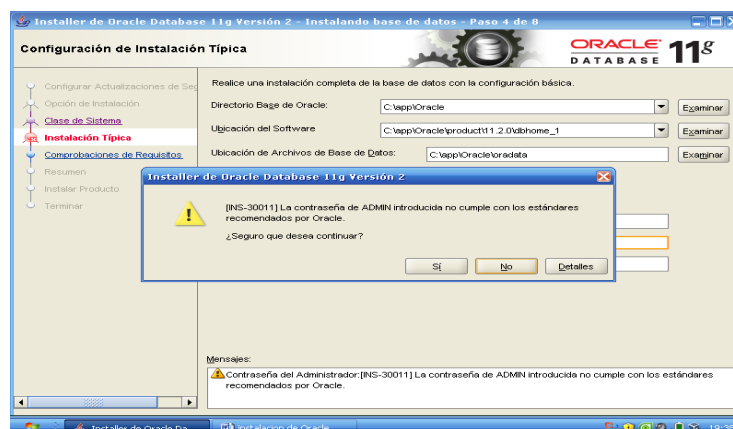


Figura 2.1.5.1.15

Afirmación de contraseña

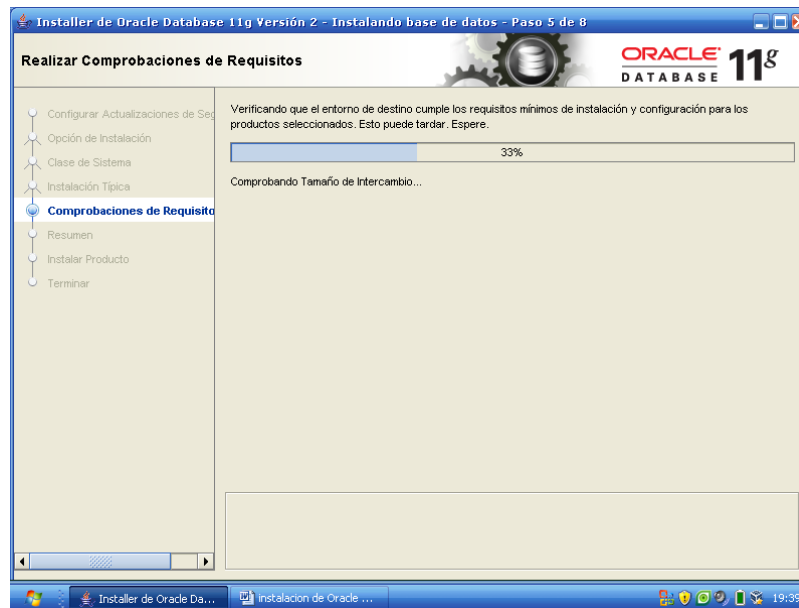


Figura 2.1.5.1.16

Posteriormente, se iniciara la comprobación de requisitos para Oracle.

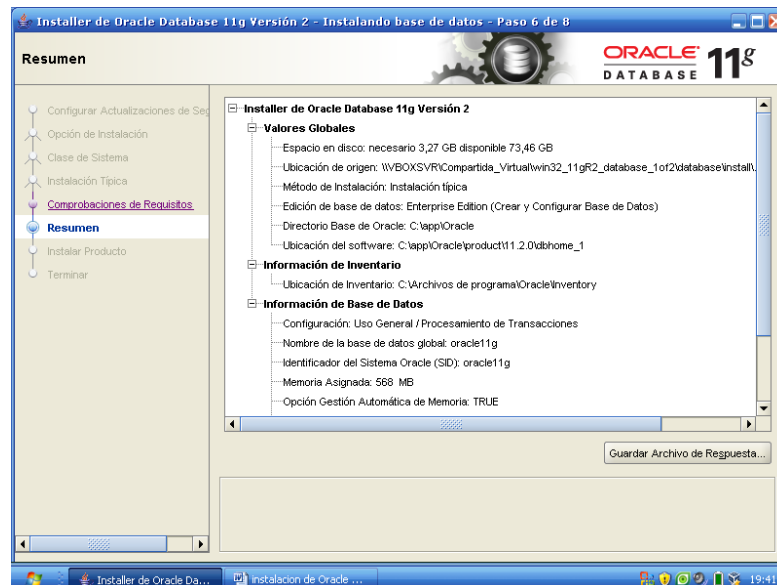


Figura 2.1.5.1.17

Se mostrara un resumen de la información

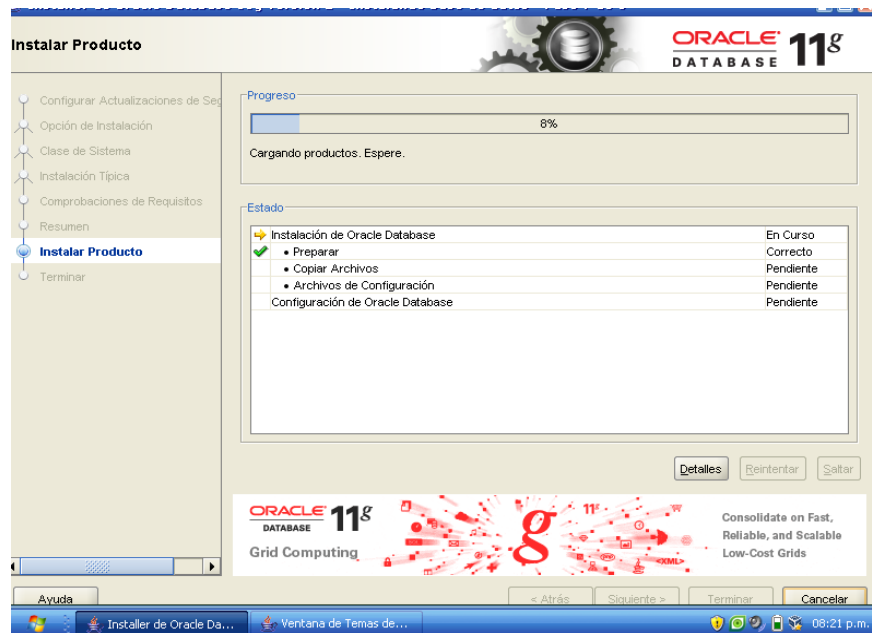


Figura 2.1.5.1.18

Por último, se iniciara la instalación de Oracle, esto puede tardar varios minutos definiéndose, según las características de la computadora.

2.2 Recopilación de Requisitos Técnicos de Java.

2.2.1 Requerimientos de hardware de Java

- Intel Pentium 4 (o x86 equivalente) a 2,8 GHz o procesador más rápido.
Procesador multinúcleo recomendado
- 4 GB de RAM
- 3 GB de espacio disponible en el disco duro
- Módulo de seguridad de hardware (HSM) recomendado

2.2.2 Requerimientos de software de Java

Es necesario contar con cierto software de desarrollo Java para la compilación y ejecución de las aplicaciones generadas.

- JRE (Entorno de ejecución)
- JDK (Entorno de desarrollo y testeo)
- JEE (Entorno para crear aplicaciones empresariales que se usen tecnología de servidores)
- JME (Entorno de desarrollo para móviles)
- Driver JDBC
- Navegador
- En caso de aplicaciones web (servlets), adicionalmente se requiere:
 - Un Servidor Web
 - Un Motor de Servlets
 - JavaServer Web Development Kit
- En caso de aplicaciones 3 capas, dependiendo del protocolo, adicionalmente se requiere:
 - Si es DCOM – no se requiere SW adicional
 - Si es HTTP – se requiere un servidor web y un motor de servlets
 - Si es CORBA – se requiere el Visibroker
 - Si es RMI – se requiere el rmi.zip

Applets:

Las applets de Java: son programas incrustados en otras aplicaciones, normalmente una página Web que se muestra en un navegador.

Servlets:

Los servlets son componentes de la parte del servidor de Java EE, encargados de generar respuestas a las peticiones recibidas de los clientes.

Para poder ejecutar los Servlets dentro de un servidor Web, es necesario un servidor de web que soporte Servlets, o un motor de Servlets en caso de que el servidor de Web no lo soporte.

- Aplicaciones con ventanas:

Swing es la librería para la interfaz gráfica de usuario avanzada de la plataforma Java SE.

- Drivers JDBC

El generador GeneXus Java solamente crea aplicaciones Cliente/Servidor.

Junto con el JavaServer Web Development Kit se distribuye un servidor de Servlets independiente (no se puede instalar como un componente de un servidor web existente).

- Este servidor debe ejecutarse invocando al script 'startserver' (o startserver.bat en Windows) que se encuentra en el directorio raíz de donde se instaló el producto.
- Tecleando PATH en una ventana de comandos de MS-DOS se muestran los nombres de directorios incluidos en dicha variable de entorno.

- Antes de ejecutar el script es necesario editarlo para agregar en el CLASSPATH el archivo gxclassr.zip (de modo que pueda ejecutar las aplicaciones generadas por GeneXus).

2.2.3 Generalidades de Java

QUÉ ES JAVA?:

Es un lenguaje de programación orientado a objetos. Es la tecnología subyacente que permite el uso de programas, punteros, como herramientas, juegos y aplicaciones de negocios.

JAVA ES UTILIZADO:

Para jugar en línea, participar en sesiones de chat con internautas de todo el mundo, calcular los intereses de una hipoteca y ver imágenes en tres dimensiones, entre otras muchas aplicaciones.

Es también esencial para las aplicaciones de intranet y otras soluciones de comercio electrónico que constituyen la base informática de las empresas.

2.2.4 Proceso de Instalación y configuración de Java

2.2.4.1 Instalación del jdk v.7.0 de java, para la instalación previa de NetBeans.



Figura 2.2.4.1.1

Hacer doble clic o enter, en el icono instalador de jdk de java.

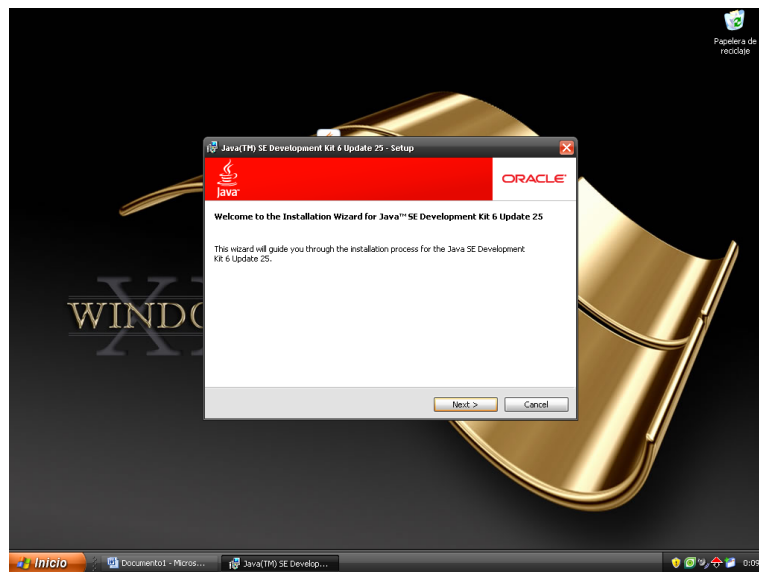


Figura 2.2.4.1.2

El asistente de instalación del JDK se iniciara.

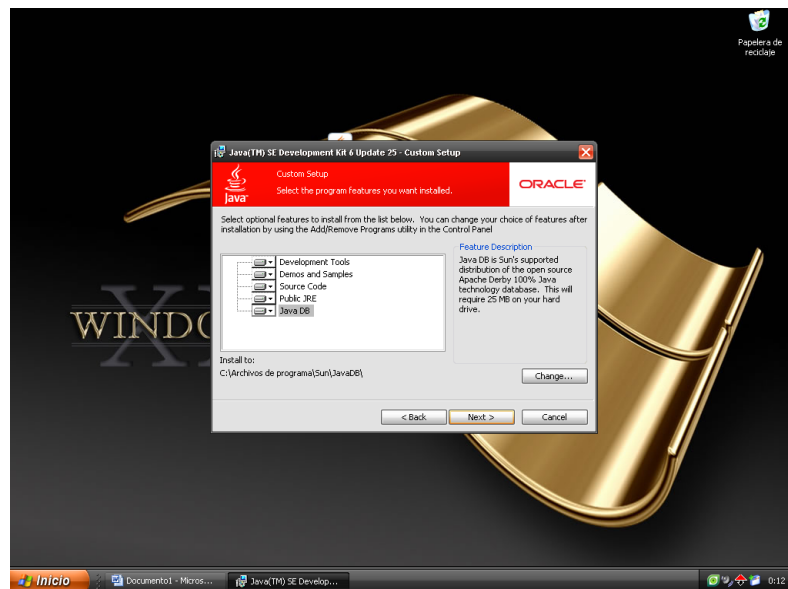


Figura 2.2.4.1.3

Pulsar siguiente, para continuar con la instalación.

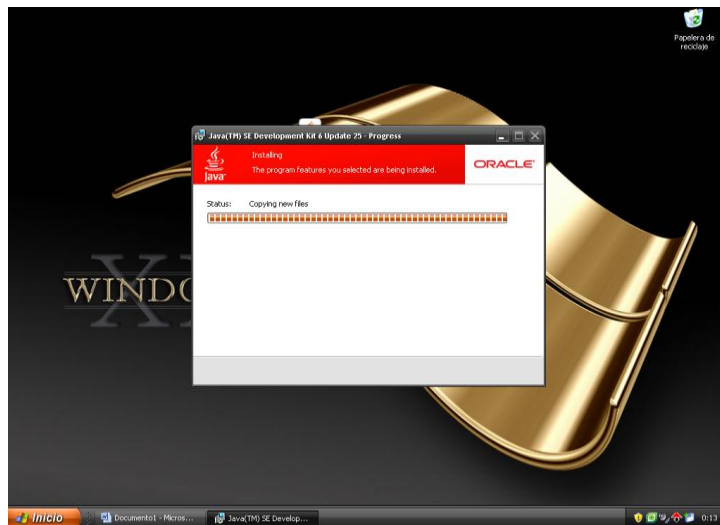


Figura 2.2.4.1.4

La instalación, se iniciara mostrando el progreso, a través de una barra.

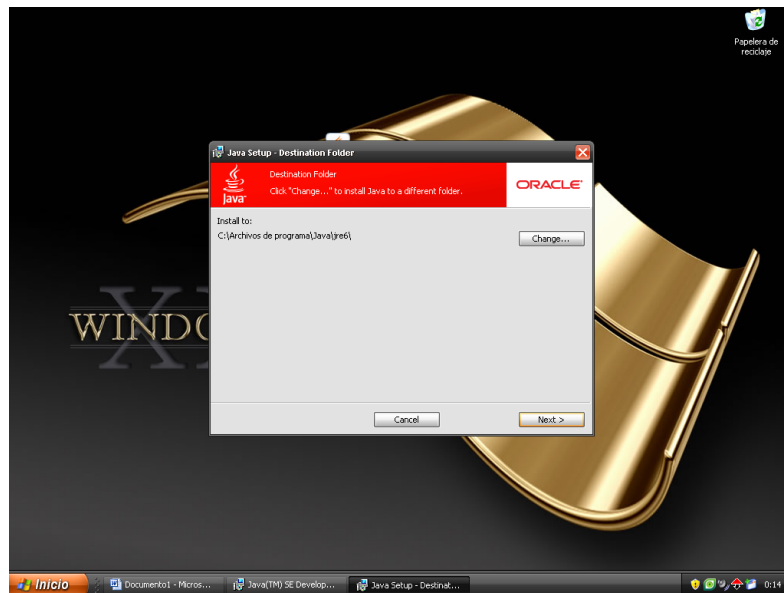


Figura 2.2.4.1.5

En este punto se podrá elegir la ruta de instalación de JDK.

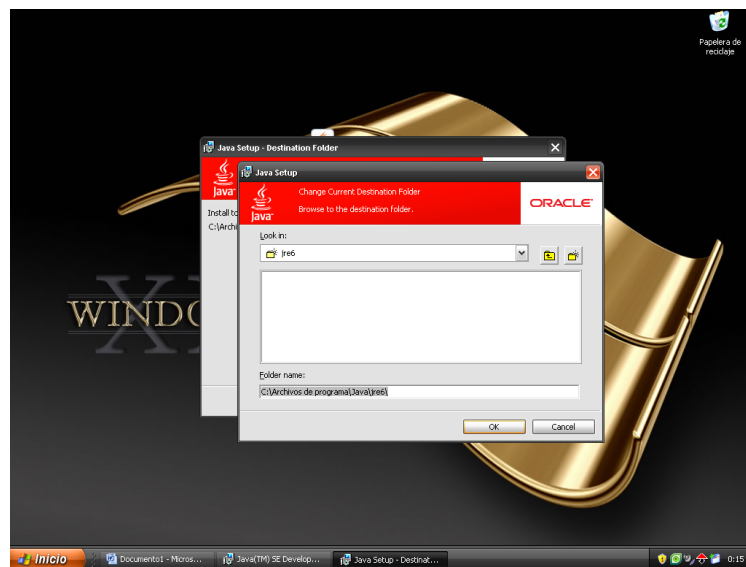


Figura 2.2.4.1.6

Hacer clic en ok, para aceptar la ruta de instalación.



Figura 2.2.4.1.7

La instalación continuara registrándose, en la carpeta de registro elegida.

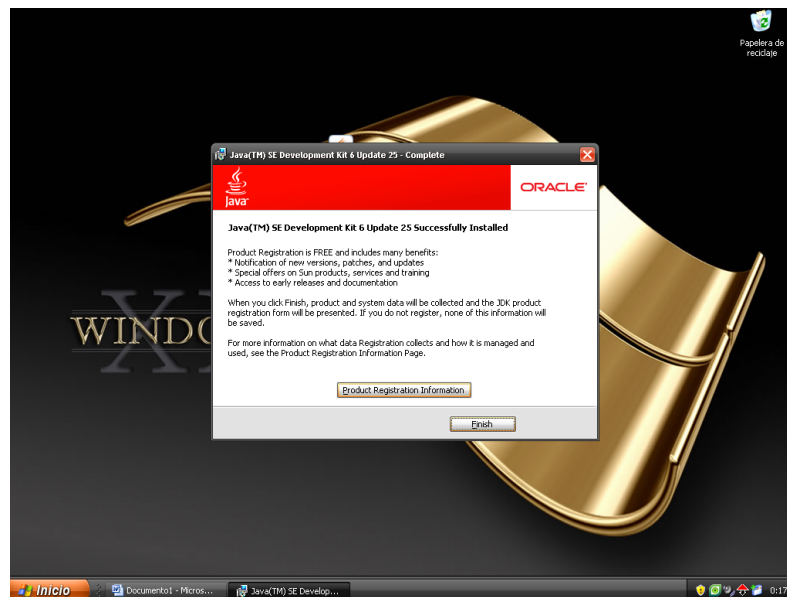


Figura 2.2.4.1.8

Muestra el registro de la instalación.

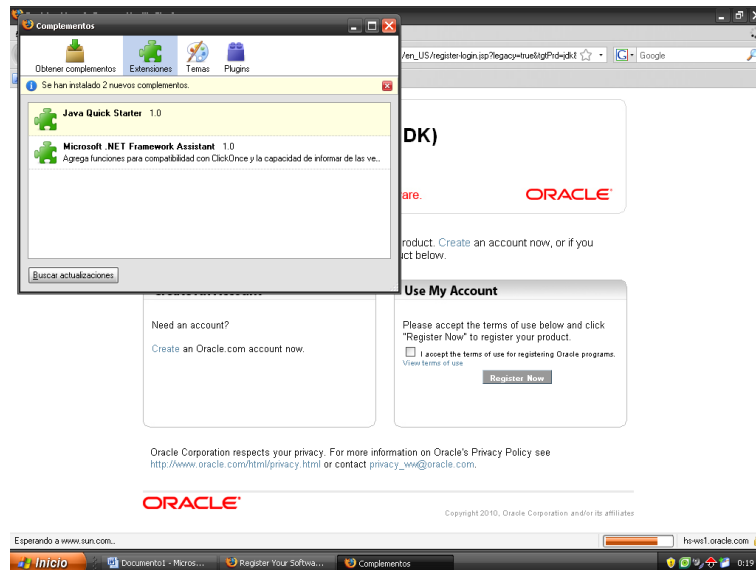


Figura 2.2.4.1.9

Se cargara la página web, del registro de JDK

Se cargara la página web de jdk de java, para actualizaciones.



Figura 2.2.4.1.10

Registro de jdk

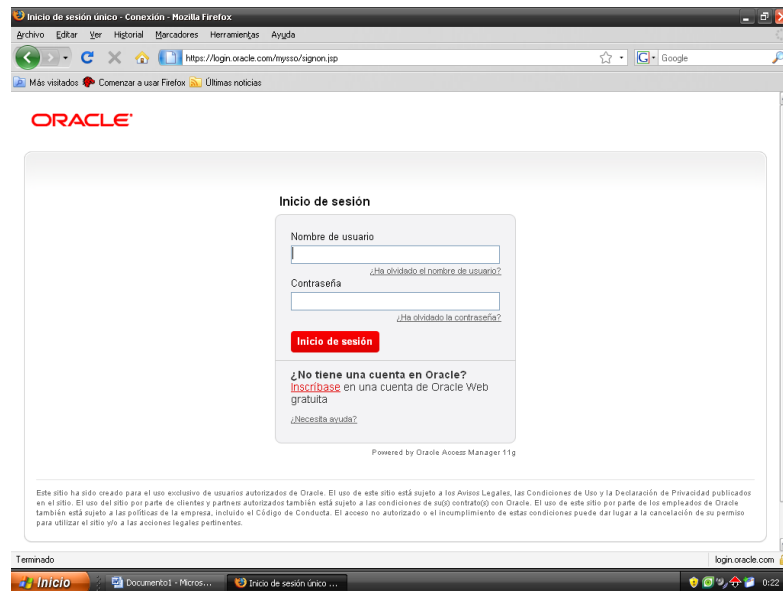


Figura 2.2.4.1.11

Ingreso de información requerida.

2.3 Recopilación de Requisitos de NetBeans.

2.3.1 Requerimientos de Hardware de NetBeans.

Microsoft Windows XP Professional SP2:

Processor: 2.6 GHz Intel Pentium IV or equivalent

Memory: 1 GB

Disk space: 1 GB of free disk space

Microsoft Windows xp:

Processor: 2.6 GHz Intel Pentium IV or equivalent

Memory: 1 GB

Disk space: 1 GB of free disk space

2.3.2 Generalidades del NetBeans.

QUE ES NETBEANS: es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java.

PARA QUE SIRVE: como una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación.

2.3.3 Requisitos de Instalación de NetBeans.

- Java SE. Proporciona todas las características estándar para el desarrollo de Java SE. También se incluye la compatibilidad con el desarrollo de RCP de la plataforma NetBeans.
- JavaFX. Proporciona el SDK de JavaFX 1.3.1 y el compositor de JavaFX, para el desarrollo de aplicaciones multiplataforma de JavaFX.
- Java. Proporciona herramientas para desarrollar aplicaciones de Java SE, Java EE y Java ME. También se incluye la compatibilidad con el desarrollo de RCP de la plataforma NetBeans. Esta opción de descarga también incluye el servidor GlassFish OSE (Open Source Edition) 3.0.1, y Apache Tomcat 6.0.26.

- Ruby. Proporciona herramientas para el desarrollo de Ruby y es compatible con Rails y JRuby. También incluye el servidor GlassFish OSE (Open Source Edition) 3.0.1.
- C/C++. Es compatible con el desarrollo en los lenguajes C, C++, Qt, Fortran y Assembly.
- PHP. Proporciona herramientas para el desarrollo de PHP 5.x, y los Frameworks Zend y Symfony.
- All (Todo). Ésta es la opción de descarga completa, que incluye todos los entornos de ejecución y los paquetes disponibles para el IDE NetBeans.

2.3.4 Instalación y configuración de NetBeans.

2.3.4.1 Instalación de NetBeans

Se Inicia la instalación de Netbeans



Figura 2.3.4.1.1

Hacer clic, en el icono de instalación de NetBeans 7.1.

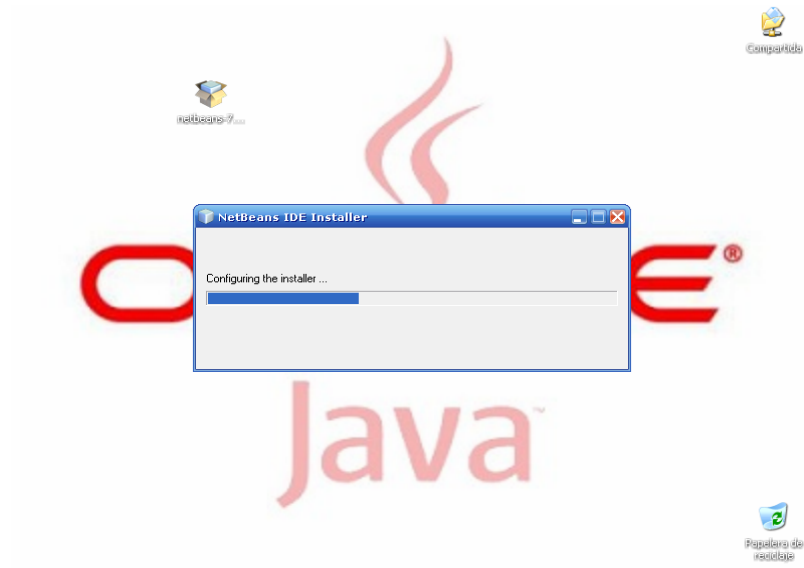


Figura 2.3.4.1.2

Inicia la carga

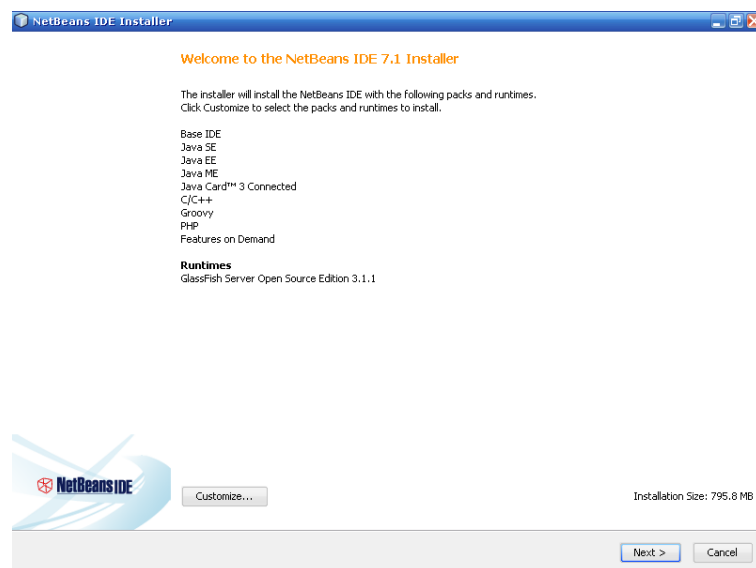


Figura 2.3.4.1.3

Pulsar siguiente, para continuar con la instalación

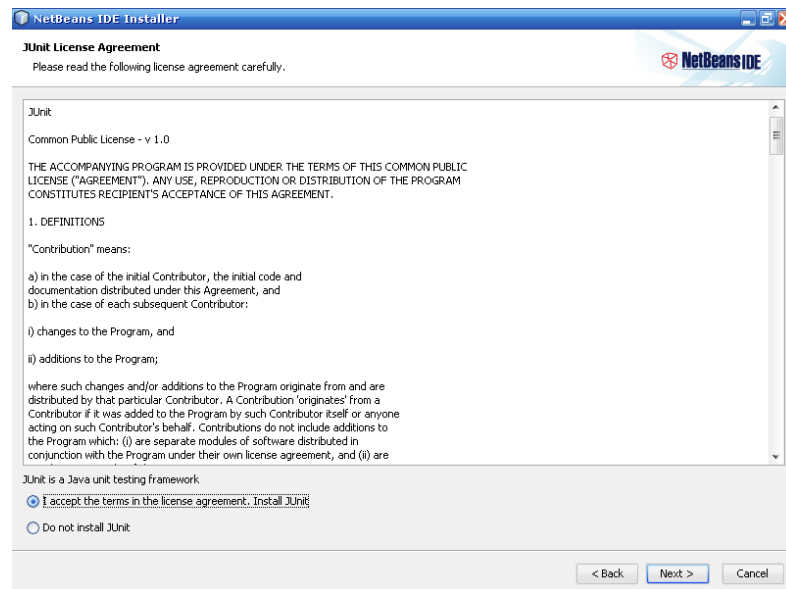


Figura 2.3.4.1.4

Aceptar los términos de licencia del NetBeans 7.1, Pulsar siguiente o cambiar la ubicación de la instalación.

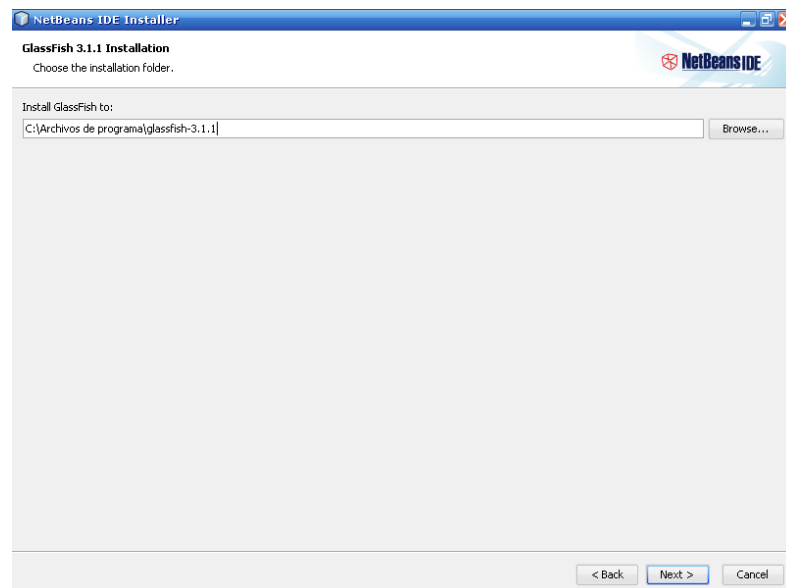


Figura 2.3.4.1.5

Elegir la ruta de instalación, de GlassFish y presionar siguiente

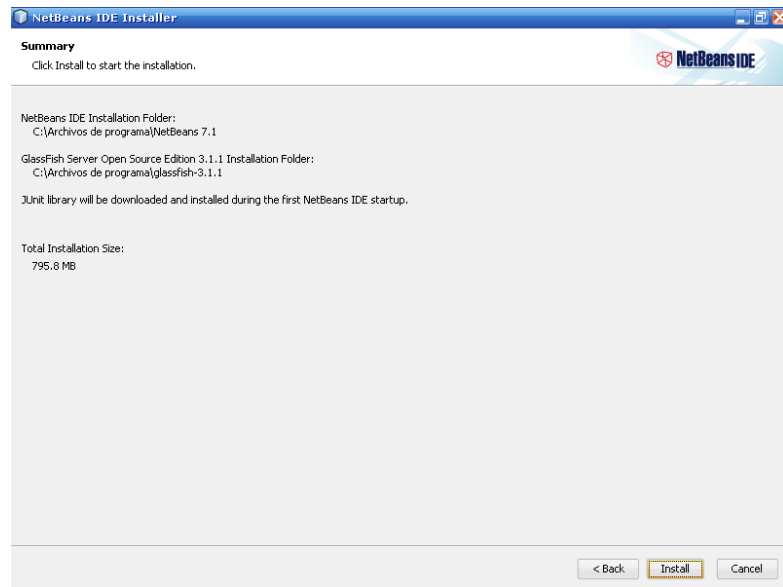


Figura 2.3.4.1.6

Hacer clic, para iniciar la instalación

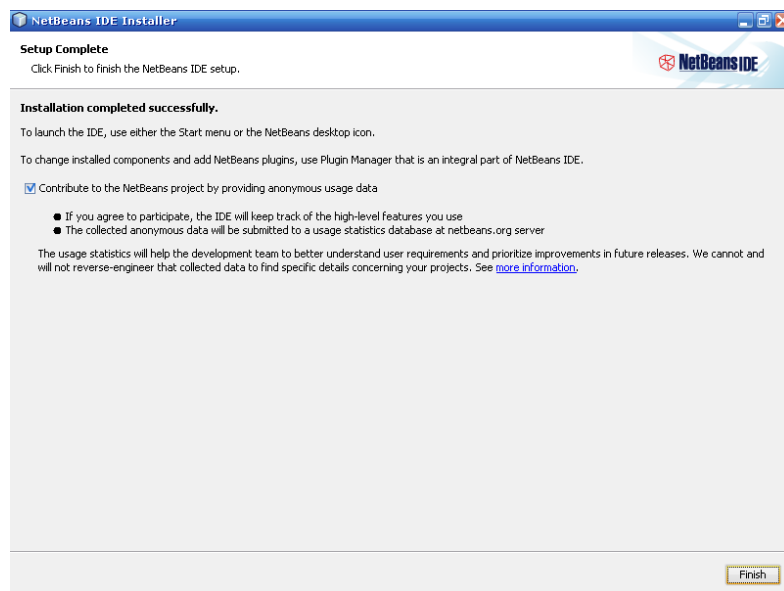


Figura 2.3.4.1.7:

Barra de progreso de la instalación del NetBeans y hacer clic en finalizar

2.4 Recopilación de Requisitos de Moodle.

2.4.1 Requerimientos de Hardware de Moodle.

- Espacio en disco: 160 MB libres (mínimo),
- Memoria: 256 MB (mínimo), 1 GB o más es muy recomendable.
- La regla general es que Moodle puede albergar de 10 a 20 usuarios simultáneos por cada 1 GB de RAM,

2.4.2 Requerimientos de software de Moodle.

- Un servidor web. La mayoría de los usuarios usan Apache, pero Moodle debe funcionar bien en cualquier servidor web que soporte PHP, como el IIS (Internet Information Server) de las plataformas Windows.
- Una instalación de PHP en funcionamiento (versión 4.3.0 o posterior). PHP 5 está soportado a partir de Moodle 1.4. (tener cuidado con PHP-Accelerator ya que se han detectado problemas con él. Moodle 2 necesita como mínimo la versión 5.2.8. Si se usa PHP 5.3, hay que asegurarse que la versión es superior a la 5.3.2
- Una base de datos: MySQL o PostgreSQL, que están completamente soportadas y recomendadas para su uso con Moodle.
- MySQL al utilizar la base de datos MySQL. En algunas distribuciones de Linux (principalmente RedHat) se trata de un paquete opcional.

Requerimientos adicionales:

- Librería GD y librería FreeType 2 para poder construir los gráficos de los registros de Moodle.
- mbstring - es requerido para manipular cadenas de caracteres multi-byte (iconv también es recomendable para Moodle 1.6).

La extensión zlib es necesaria si se va a utilizar las funcionalidades zip/unzip.

- GNU/Linux: LAMPserver: Apache + PHP 5 + MySQL .
- Windows: WAMPserver: XAMPP o WOS.
- Descompresión de moodle.zip
- Copia en directorio/www de Apache.
- Creación de directorio de datos moodle data.
- Creación de la base de datos (desde PHPMyAdmin).

2.4.3 Generalidades del Moodle.

“MOODLE: Es un paquete de software para la creación de cursos y sitios Web basados en Internet. Es un proyecto en desarrollo diseñado para dar soporte a un marco de educación social constructivista.

2.4.4 Requisitos de Instalación de Moodle.

Ejecutar el script de instalación para crear config.php

Para ejecutar el script de instalación (install.php), se ingresa a la dirección URL de la instalación realizada de Moodle, usando un navegador web, o simplemente acceder a `http://suservidor/install.php` directamente.

Moodle detecta la configuración necesaria y a través de algunas pantallas ayuda a crear el archivo de configuración llamado, `config.php`.

Moodle escribirá el archivo en el lugar apropiado, al no ser posible, se baja desde el instalador y se coloca al directorio principal de Moodle en el servidor.

Configuración general del servidor web

- Primero se establece la configuración en el servidor web, para usar `index.php`, como página inicial (quizá además de `index.html`, `default.htm`, etc.).

En Apache, se usa un parámetro, `DirectoryIndex` en el archivo `httpd.conf`. Habitualmente aparece de esta forma:

```
DirectoryIndex index.php index.html index.htm
```

- Segundo, si se utiliza Apache 2, se debe de habilitar la variable `AcceptPathInfo`, la cual permite que se puedan pasar argumentos a los scripts como `http://servidor/archivo.php/arg1/arg2`.

```
AcceptPathInfo on
```

- Tercero, Moodle necesita tener activada, una determinada configuración en la instalación PHP, para funcionar.

2.5 Cronograma de Actividades.

ACTIVIDAD	feb-12				mar-12					abr-12				may-12				jun-12			
	4	11	18	25	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23
Asignacion de grupo y Tema																					
Desarrollo del Capitulo I																					
Situacion Actual																					
Entrega del primer avance																					
Desarrollo del Capitulo II																					
Documentacion Tecnica																					
Entrega del segundo avance																					
Retiro del primer avance																					
Desarrollo del Capitulo III																					
Diseño de la Solucion																					
Retiro del segundo avance																					
Entrega del tercer avance																					
Retiro del tercer avance																					
Configuracion de la bases de datos																					
finalizacion del proyecto																					
Entrega del trabajo final																					

2.6 Presupuesto

El recurso económico para la elaboración del proyecto proviene de fondos personales.

DETALLE	CANTIDAD	TOTAL
5 Cuotas de Seminario de graduación	\$ 57.14	\$ 571.40
Examen de seminario de graduacion	\$ 85.71	\$ 171.42
Transporte	12	\$ 65.00
Material e impresiones	7	\$ 70.00
Alimentacion y documentacion requerida	4	\$ 68.00
		\$ 945.82

CAPITULO III. DISEÑO DE LA SOLUCION

3.1 Diseño de la Solución

En esta etapa se describen guías con ejercicios prácticos para el uso de la creación de las bases de datos; el cual será útil en la elaboración del material de apoyo para docentes y estudiantes de la Escuela de Informática de la UTEC.

Diseño del material didáctico: Se refiere al diseño del material que será utilizado como apoyo didáctico, para las creaciones de las bases de datos en Oracle y Java.

Instalación y configuración del manual didáctico: Se muestran las pantallas de los procesos paso a paso para la instalación del software.

Diseño de Ejercicios prácticos: Se describen ejemplos de consultas de bases de datos.

Diseño de Ejercicios propuestos: Se detallan guías para el desarrollo del conocimiento básico sobre la creación de las bases de datos.

3.2 Diseño del Material Didáctico

Es el diseño del material didáctico, que será utilizado como parte del material de apoyo para docentes y estudiantes, sobre Oracle; en el cual se detallaran las funciones de configuración de la base de datos que será creada en Oracle, conteniendo lineamientos, sentencias SQL, ejemplos prácticos, entre otros,

convirtiéndose en un instrumento guía con los pasos detallados desde cómo se instala hasta como se configura la base de datos.

El usuario desarrolla habilidades relacionadas a las instalaciones de los software, configuraciones de las bases de datos, la utilización de la construcción de las base de datos, con especial énfasis en Oracle, Java y el uso del aula virtual en la plataforma Moodle.

Competencias instrumentales: Es la utilización de la ingeniería de software para implementaciones de cambios de componentes e infraestructura, que conllevan a la validación, verificación e inspección del código de las estructuras de las bases de datos.

Competencias interpersonales: Ofrecer los manuales con información puntual de lo que los estudiantes buscan o necesitan; en el momento oportuno para aprender sobre bases de datos.

Competencias sistémicas: Convertir la herramienta del aula virtual para ser utilizado en esencial para la formación de los profesionales de la UTEC en la Escuela de Informática.

Para el diseño y ejecución del Material se requiere tener en cuenta los siguientes aspectos:

- **Objetivo:** Que el alumno obtenga los conocimientos de los principios básicos de la administración y programación de la base de datos; Contando con una base de datos reconocida como Oracle, que soporte

el desarrollo de aplicaciones, en una variedad de herramientas y que una vez que la aplicación; este en producción nos brinde la seguridad de que los trabajos realizados por la base de datos, están 100% garantizados.

- **Instalación de aula virtual en la plataforma de moodle:** Esta herramienta servirá para el aprendizaje on-line; para que el alumno interactúe con su maestro ex aula, consulte con compañeros de la sección o dejar comentarios sobre el tema.
- **Guías de ejercicios de sentencias SQL:** Basadas en inserción de datos, eliminación de datos, creación de tablas.
- **Manuales de instalación y configuración de software Oracle 11g, Netbeans, Java:** Una guía práctica ilustrada para la instalación.

Para la elaboración y disposición del Material de apoyo didáctico se recomienda tener en cuenta los recursos tecnológicos y los documentos publicados en el aula virtual, deben estar en constante revisión, debido a la dinámica de los avances tecnológicos.

3.3 Instalación y Configuración

3.3.1 Guía de la Instalación y Configuración de la Base de Datos

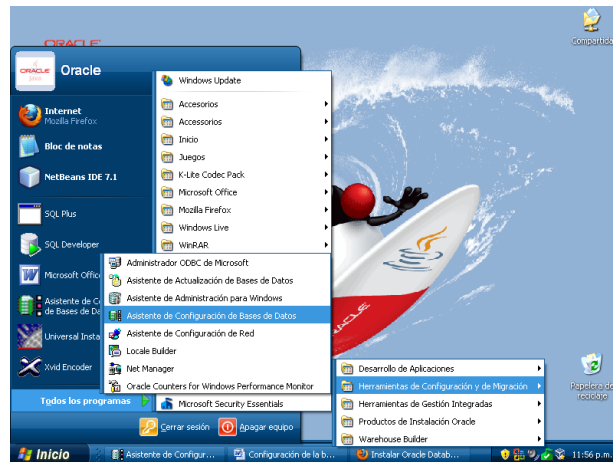


Figura 3.3.1.1.1

Hacer clic en el botón "Inicio", "Todos los programas", "Oracle OraDb11g_home1" (o el nombre que le hayamos dado a la instalación) "Herramientas de Configuración y de Migración", "Asistente de Configuración de Bases de Datos.

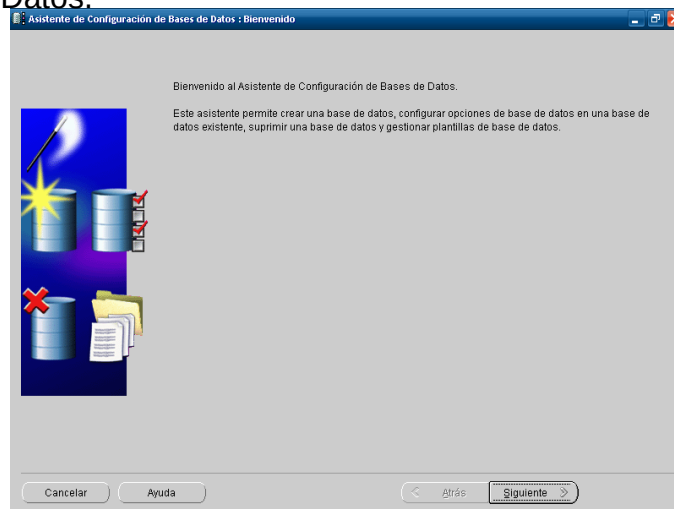


Figura 3.3.1.1.2

Esta figura muestra el inicio del Asistente de Configuración de Bases de Datos, Hacer clic en "Siguiente".

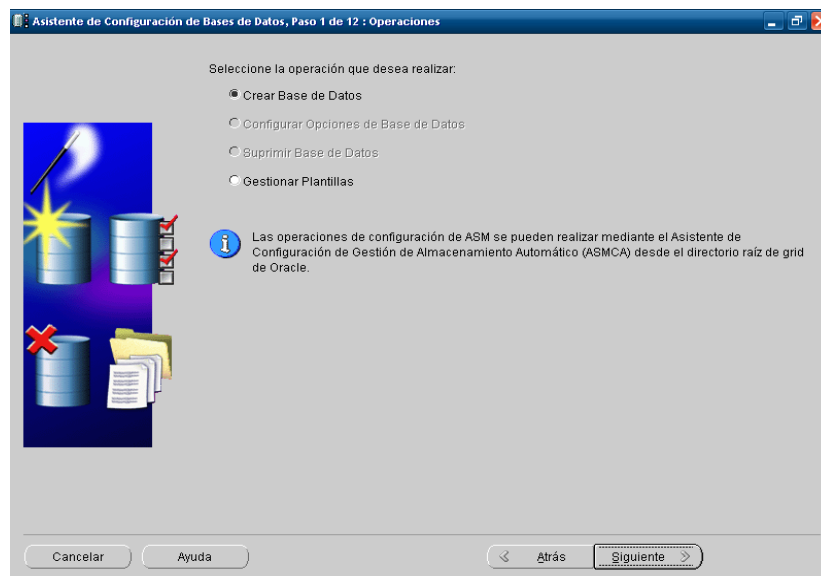


Figura 3.3.1.1.3

Marcar la opción "Crear Base de datos" y pulsaremos "Siguiente".

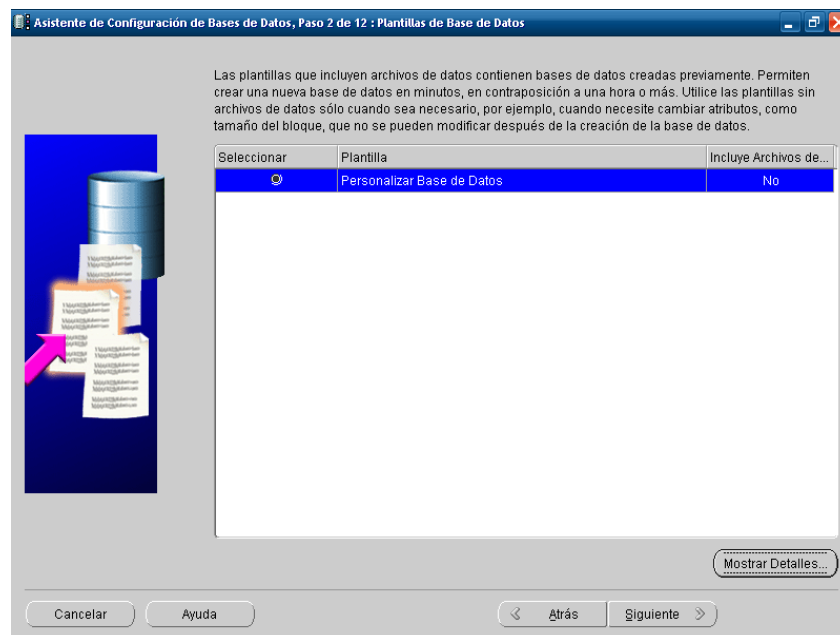


Figura 3.3.1.1.4

En este punto pulsaremos clic en "Siguiente".

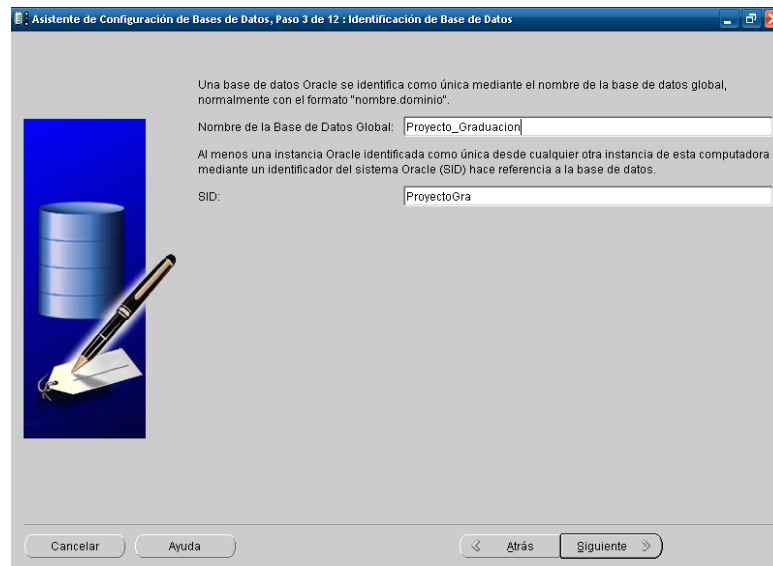


Figura 3.3.1.1.5

Se nombra la base de datos y hacer clic en siguiente.

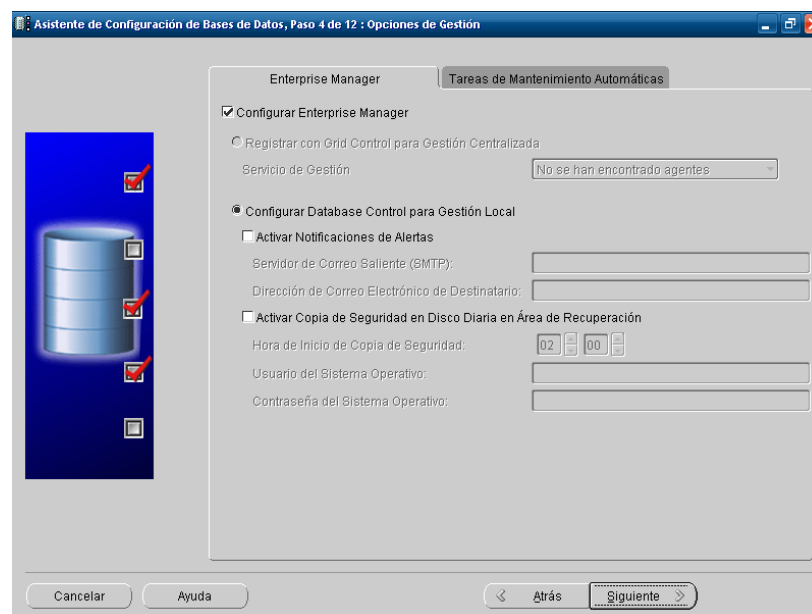


Figura 3.3.1.1.6

A continuación podremos indicar los siguientes datos:

- Configurar Enterprise Manager: proporciona herramientas de gestión basadas en Web, para gestionar instancias de base de datos.
- Configurar Database Control para Gestión Local: permite gestionar la base de datos.
- Activar Notificaciones de Alertas: si desea que el usuario SYSMAN (el súper administrador por defecto) reciba notificaciones de alerta cuando una métrica para una condición especificada alcance un umbral crítico o de advertencia.
- Activar Copia de Seguridad en Disco Diaria en Área de Recuperación.

Luego de realizar las marcas, hacer clic en siguiente.

Asistente de Configuración de Bases de Datos, Paso 5 de 12 : Credenciales de Base de Datos

Por motivos de seguridad, debe especificar contraseñas para las siguientes cuentas de usuario en la nueva base de datos.

☐ Usar Contraseñas Administrativas Diferentes

Usuario	Contraseña	Confirmar Contraseña
SYS		
SYSTEM		
DBSNMP		
SYSMAN		

☒ Usar la Misma Contraseña Administrativa para Todas las Cuentas

Contraseña:

Confirmar Contraseña:

Cancelar Ayuda Atrás Siguiente

Figura 3.3.1.1.7

En el siguiente paso se indica la contraseña, para que los usuarios del Asistente de Configuración de Base de Datos, creará: SYS, SYSTEM, DBNSMP y

SYSMAN. Podremos indicar una contraseña para cada usuario marcando "Usar Contraseñas Administrativas Diferentes" o la misma contraseña para todos los usuarios marcando, "Usar la Misma Contraseña Administrativa para Todas las Cuentas", y hacer clic en siguiente.

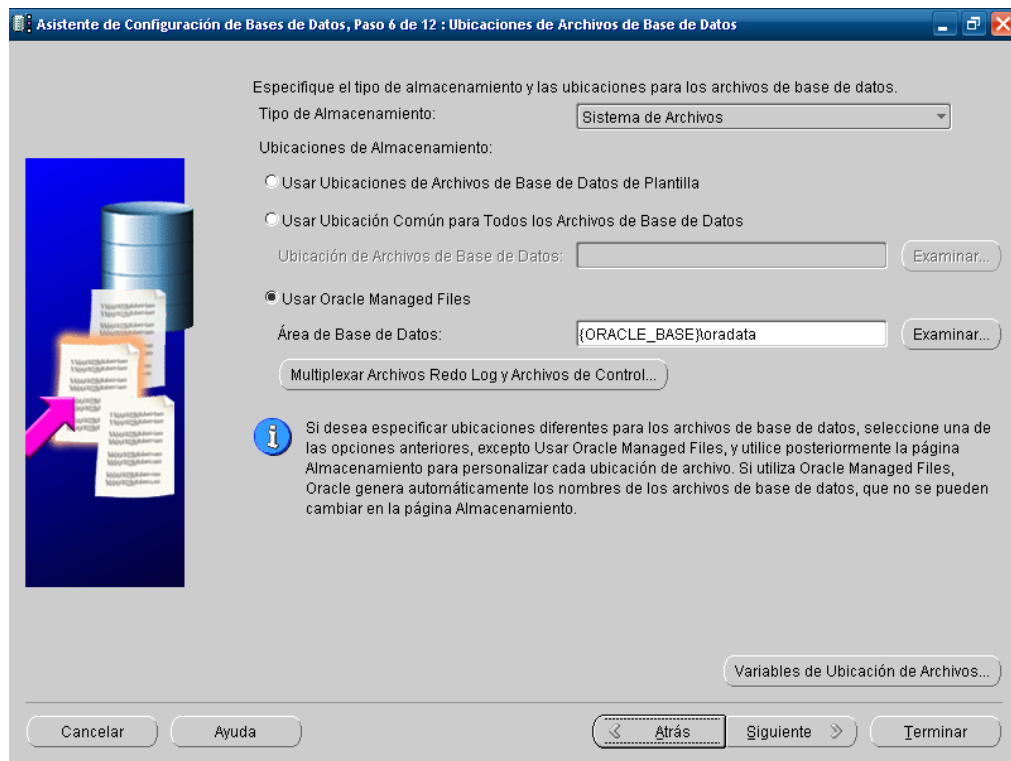


Figura 3.3.1.1.8

En esta ocasión selecciona el método de ubicaciones de archivos de base de datos Oracle Managed Files; por lo que se marcará la opción, "Usar Oracle Managed Files"

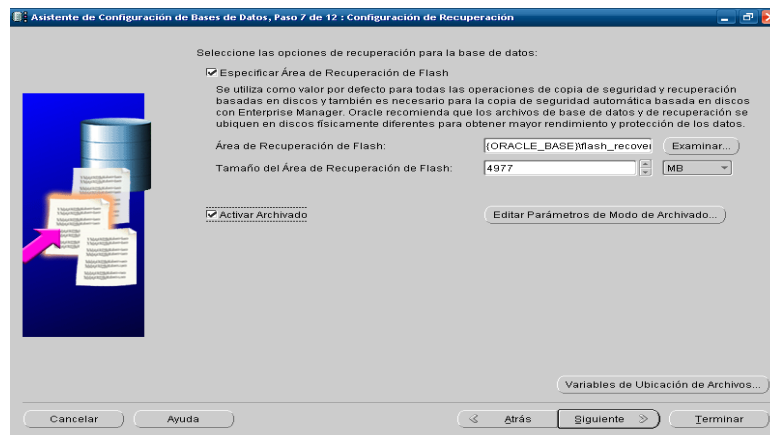


Figura 3.3.1.1.9

En la ventana siguiente podremos configurar las opciones de recuperación:

Especificar Área de Recuperación de Flash: se puede utilizar para recuperar datos que se perderían, durante un fallo del sistema.

Área de Recuperación de Flash: desde este campo se indica la ubicación de los ficheros para que Oracle gestione aquí el área de recuperación de flash.

Tamaño del Área de Recuperación de Flash: en este campo se indica el tamaño del área de recuperación de flash.

Activar Archivado: Los archivos redo log archivados se pueden utilizar para recuperar una base de datos, actualizar una base de datos estándar o recuperar información sobre el historial de una base de datos con la utilidad LogMiner.

A continuación se indica, que para tener una base de datos con alta tolerancia a fallos y por lo tanto alta disponibilidad se marca ambas opciones y luego hacer clic en siguiente.

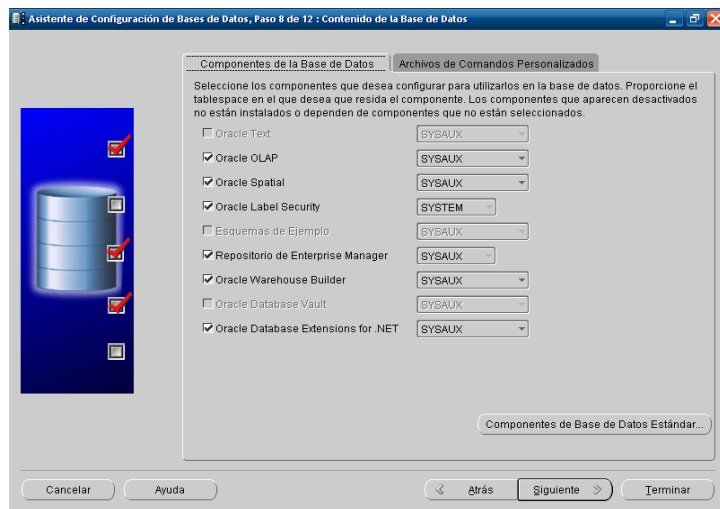


Figura 3.3.1.1.10

En la siguiente figura hacer clic en “Siguiete”.

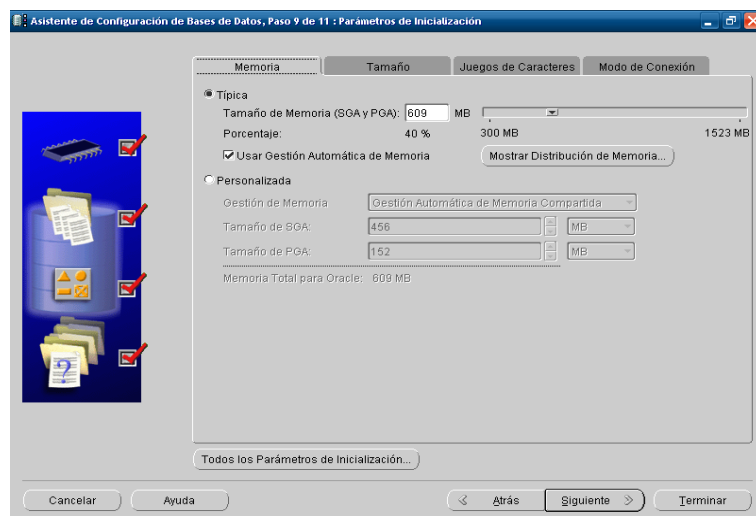


Figura 3.3.1.1.11

En el siguiente paso de la creación de la base de datos, se indican los parámetros de "Memoria", "Tamaño", "Juego de Caracteres" y "Modo de Conexión". En la pestaña "Memoria" podremos indicar los siguientes parámetros:

Típica: seleccionando esta opción se creará una base de datos con la mínima configuración o intervención del usuario. Esta opción es suficiente para la mayoría de los entornos y para los administradores de bases de datos Oracle.

Personalizada: uso de la gestión de memoria personalizada. Si el usuario es un administrador de base de datos experimentado y desea tener mayor control sobre la asignación de memoria de la base de datos Oracle, es conveniente seleccionar "Personalizada". Para asignar cantidades específicas de memoria al SGA y al PGA, seleccione Gestión Automática de Memoria Compartida para la opción Gestión de Memoria y especifique los valores adecuados.

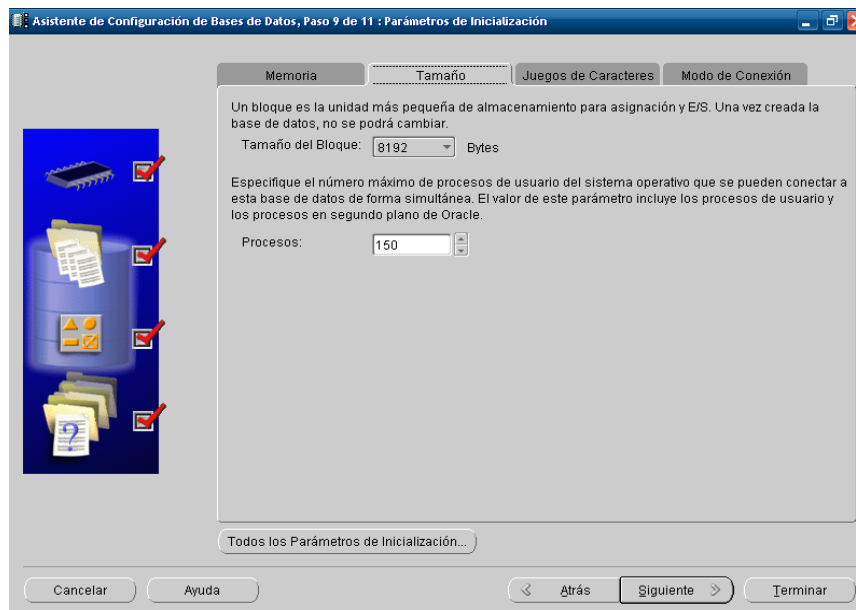


Figura 3.3.1.1.12

En la pestaña "Tamaño" definiremos los parámetros de tamaño, que se describen en la tabla siguiente:

Tamaño del Bloque: se introduce el tamaño en bytes de los bloques de la base de datos Oracle o dejaremos la opción por defecto.

Procesos: especifica el número máximo de procesos de usuario del sistema operativo que se pueden conectar simultáneamente a una base de datos Oracle. Este valor debe permitir todos los procesos en segundo plano, como bloqueos, procesos de cola de trabajos y procesos de ejecución paralela.

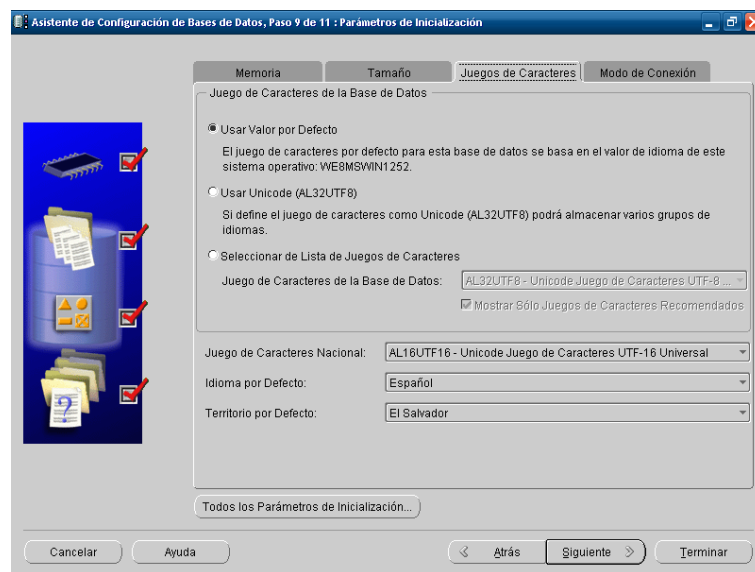


Figura 3.3.1.1.13

En la pestaña "Juegos de Caracteres", definiremos los juegos de caracteres utilizados, por la base de datos Oracle. Más concretamente, se pueden definir:

- Juego de Caracteres de la Base de Datos.
- Juego de Caracteres Nacional.
- Idioma por Defecto.
- Formato de Fecha por Defecto.

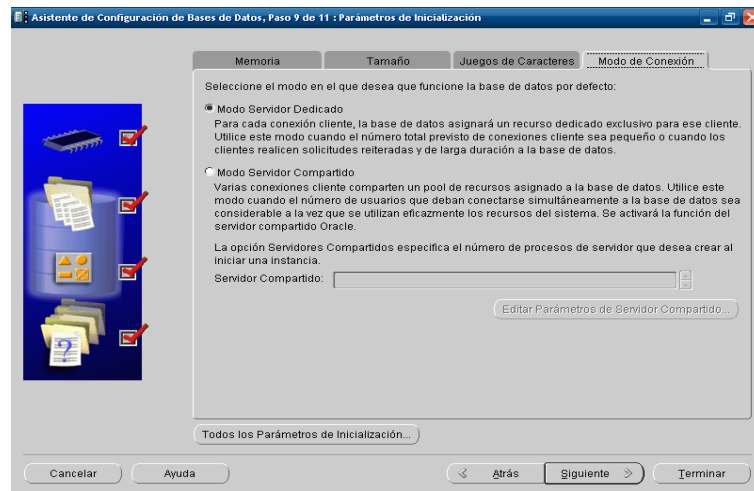


Figura 3.3.1.1.14

En la pestaña "Modo de Conexión" seleccionaremos el modo de conexión para la base de datos.

A continuación, se muestran los modos que puede seleccionar:

Modo Servidor Dedicado: necesita un proceso de servidor dedicado para cada proceso de usuario. Hay un proceso de servidor para cada cliente. La Red de Oracle devuelve la dirección de un proceso de servidor existente al cliente y éste, a su vez, vuelve a enviar la solicitud de conexión a la dirección del servidor proporcionada. Este método se seleccionará cuando:

Esté utilizando la base de datos en un entorno de almacenes de datos.

Se espera un número total de conexiones del cliente inferior.

Los clientes de la base de datos realicen solicitudes reiteradas y de larga duración a la base de datos.

Modo Servidor Compartido: una base de datos Oracle en modo Servidor Compartido (también denominado modo de servidor multithread) está configurada para que varios procesos del usuario puedan compartir un número muy reducido de procesos del servidor, de forma que aumenta la cantidad de usuarios soportados.

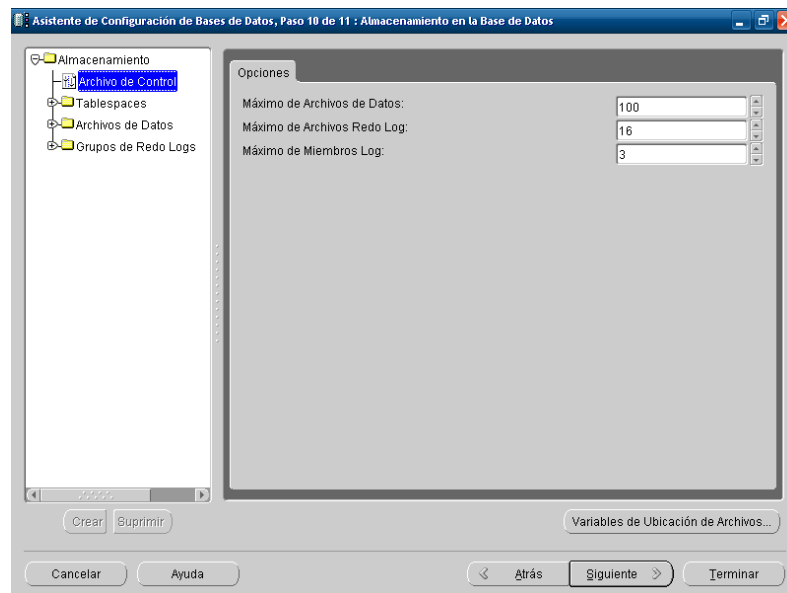


Figura 3.3.1.1.15

A continuación puede indicar las opciones de almacenamiento

Los Archivos de Control:

- Máximo de Archivos de Datos.
- Máximo de Archivos Redo Log.
- Máximo de Miembros Log.

Dejaremos las opciones por defecto.

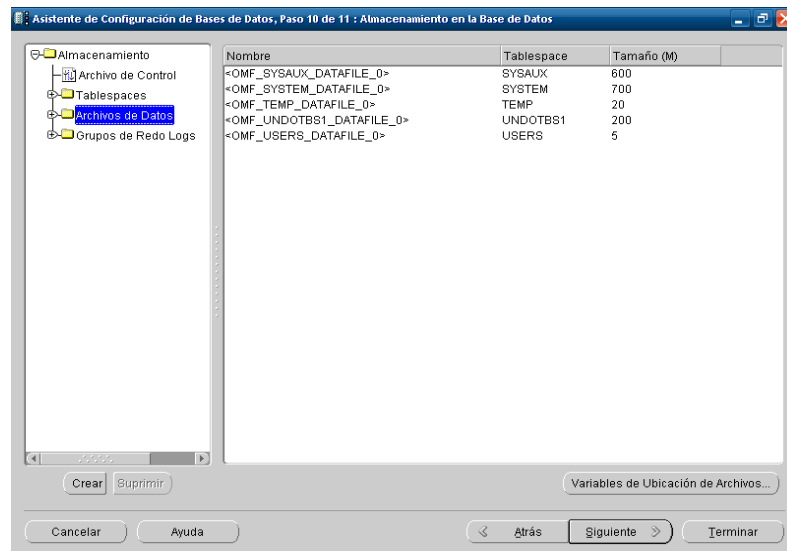


Figura 3.3.1.1.16

Archivos de Datos y presionar siguiente.

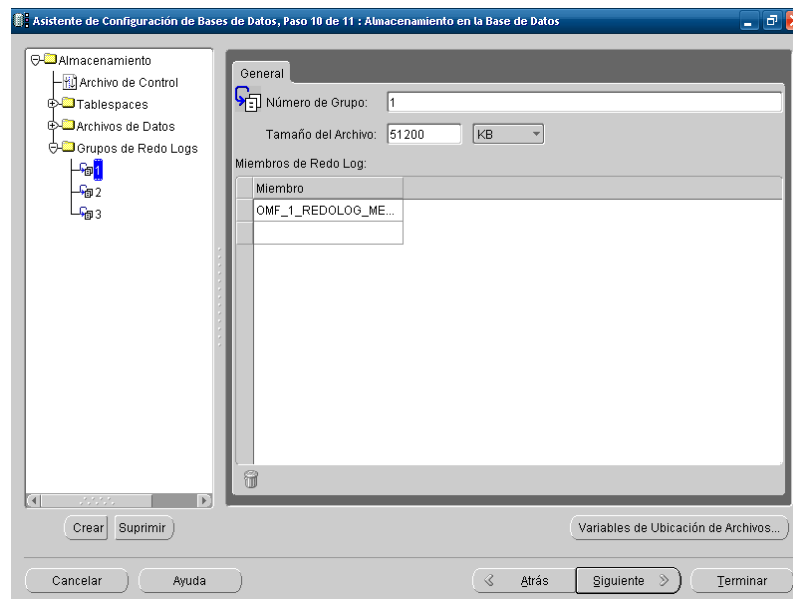


Figura 3.3.1.1.17

Grupos de Redo Logs y hacer clic en siguiente.

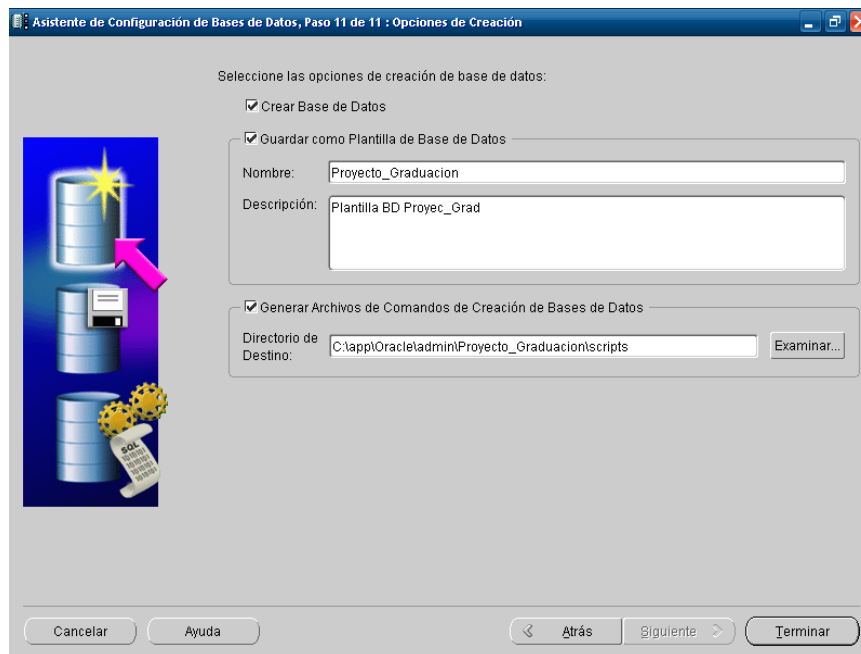


Figura 3.3.1.1.18

Antes de crear la base de datos, el asistente nos da las siguientes posibilidades:

- Crear Base de Datos: active esta casilla de control para crear la base de datos ahora.
- Guardar como Plantilla de Base de Datos: active esta casilla de control para guardar los parámetros de creación de la base de datos como plantilla.

Nombre: introduzca un título para esta plantilla.

Descripción: proporcione una descripción breve del tipo de plantilla.

Para confirmar la creación, presionar terminar.

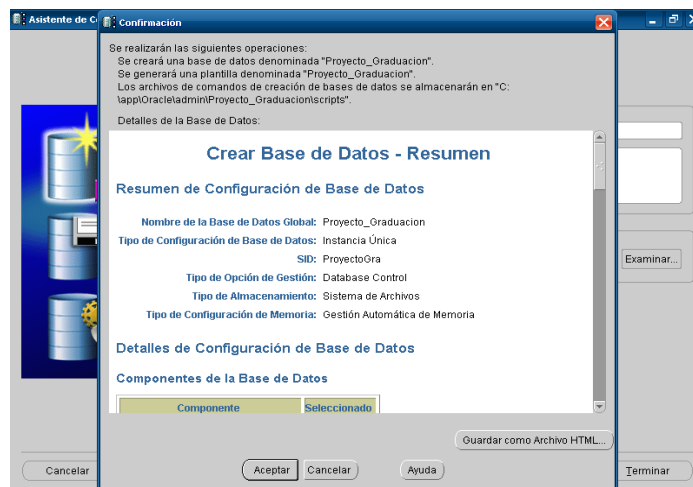


Figura 3.3.1.1.19

Después de pulsar en "Terminar" en la ventana anterior podremos ver, todas las opciones seleccionadas. Pulsaremos "Aceptar" para iniciar la creación definitiva de la base de datos Oracle

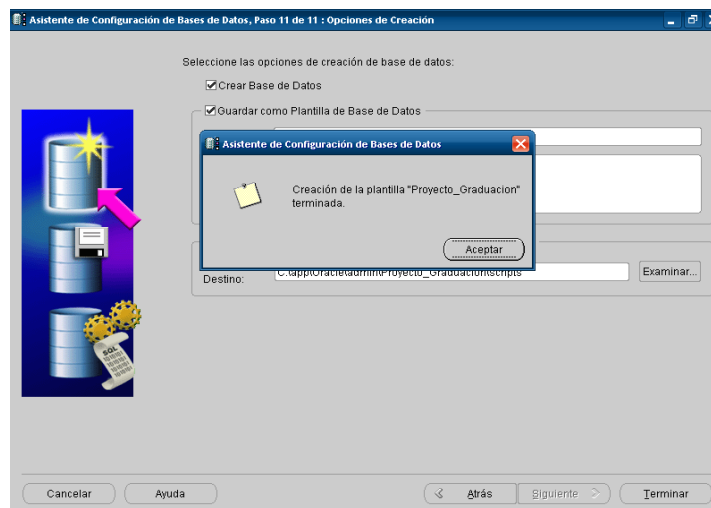


Figura 3.3.1.1.20

Nos mostrará un mensaje indicando que la plantilla se ha creado correctamente. Pulsaremos "Aceptar":

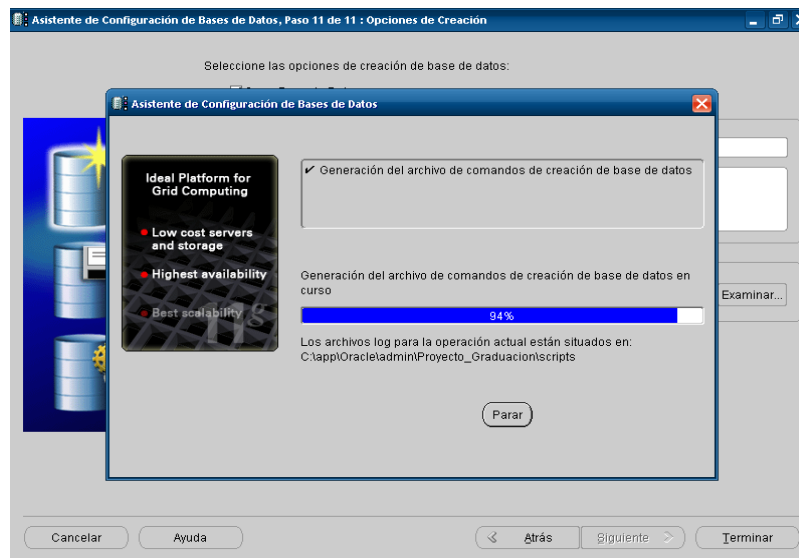


Figura 3.3.1.1.21

Se creará el archivo de comandos (si este ha sido marcado).

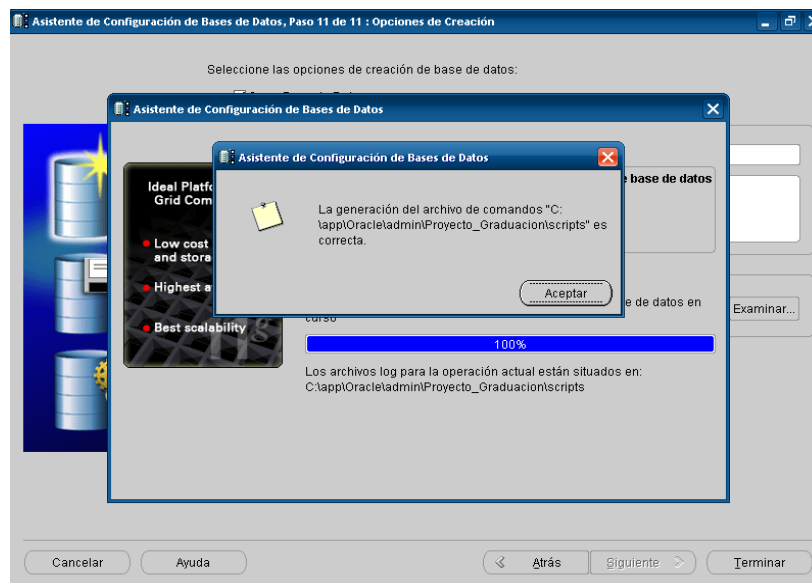


Figura 3.3.1.1.22

Mostrará un mensaje indicando que el archivo de comandos se ha creado correctamente:

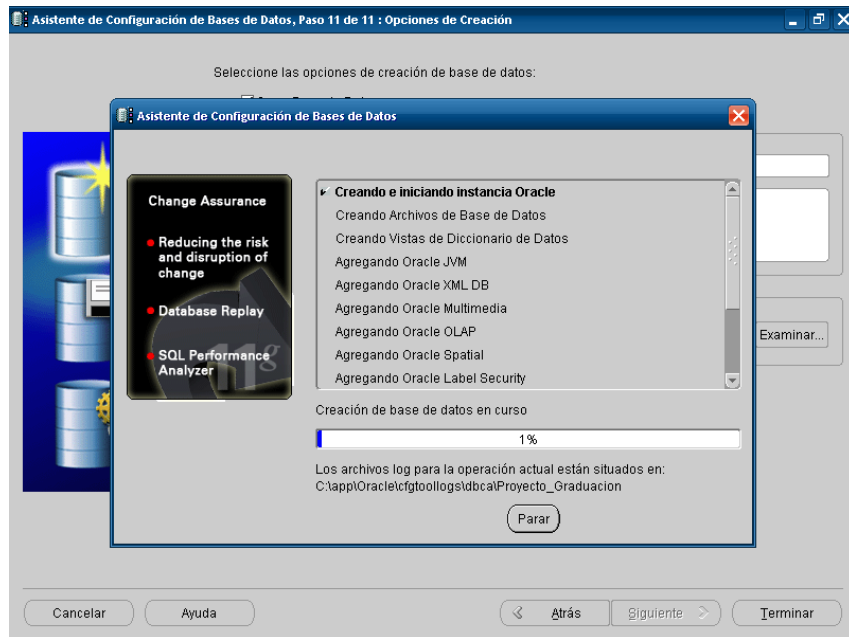


Figura 3.3.1.1.23

Se iniciará la creación definitiva de la base de datos Oracle

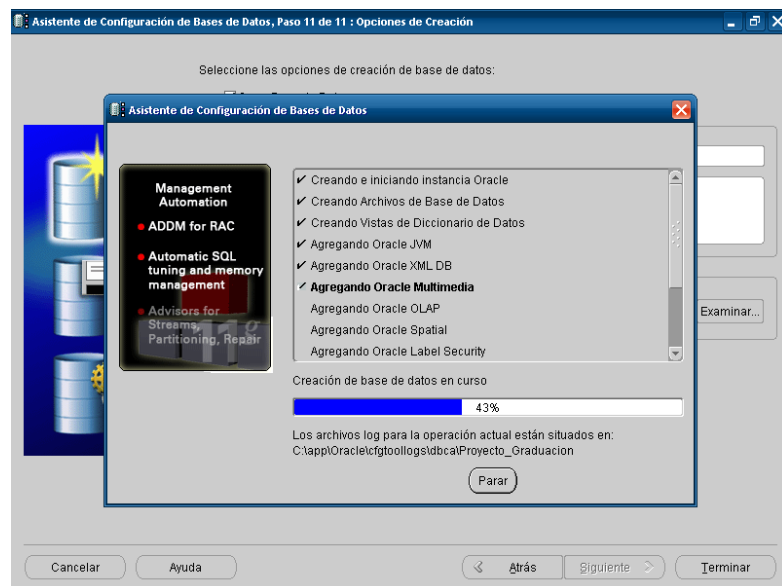


Figura 3.3.1.1.24

Progreso del proceso.



Figura 3.3.1.1.25

Fin del proceso de la creación de la base de datos y hacer clic en salir.

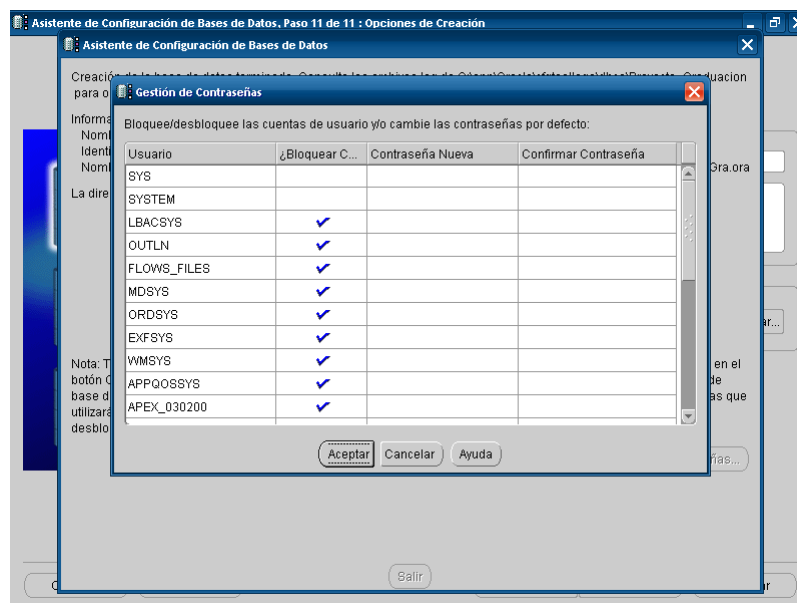


Figura 3.3.1.1.26

En este punto seleccione los usuarios que se desean bloquear y presionar aceptar.

3.4 Diseño de Ejemplos Prácticos de Oracle

Las siguientes guías que se presentan en el documento son diseñadas para cumplir las competencias a desarrollar por el estudiante.

Las guías serán desarrolladas siguiendo paso a paso las indicaciones que se le darán en cada guía para lograr el cumplimiento de los objetivos de cada una de las guías a desarrollar.

Las guías a desarrollar se mostrarán cada una de ellas en los anexos y están nombradas de la siguiente manera.

Consultas de base de datos ORACLE

Manipulación de base de datos ORACLE.

Diseño de base de datos.

Manipulación de base de datos.

Configuración de Listener.

3.5 Diseño de Ejercicios Propuestos de Java.

La siguiente guía que se presentan en el documento es diseñada para cumplir las competencias a desarrollar por el estudiante.

La guía será desarrollada siguiendo paso a paso las indicaciones que se le darán en la guía para lograr el cumplimiento de los objetivos de la guía a desarrollar.

La guía a desarrollar se mostrará en los anexos y está nombrada de la siguiente manera.

Uso de sentencias condicionales de Java.

CONCLUSION

En la elaboración del presente trabajo, llegamos a la conclusión que el desarrollo del material de apoyo didáctico sobre la base de datos Oracle 11g y el lenguaje de programación Java; el cual por medio de las figuras, se muestra el proceso de instalación, configuración de las bases de datos y la realización de los ejemplos prácticos, con la finalidad que los estudiantes puedan aprender a trabajar en estas dos tecnologías, que le serán de utilidad para ejecutar trabajos importantes dentro de una empresa.

También a través de un aula virtual utilizando moodle, podrán desarrollar los estudiantes la competencia, habilidades, aptitudes y conocimiento para el desarrollo de sistemas, creación y administración de bases de datos en el ámbito laboral.

La interacción entre el aula y el estudiante se lograra través del incentivo que este tendrá al poder participar en foros, chat, documento para desarrollar ejercicios práctico, laboratorios, guías de ejercicio propuestos.

A la vez se pretende lograr la creación de un apoyo de los temas a desarrollar en las horas clases sin necesidad de tener un docente presente físicamente.

RECOMENDACIÓN

Se recomienda a las autoridades de la Universidad Tecnológica de El Salvador a la Escuela de informática, en promover a los estudiantes, la cultura de apoyarse en fomentar el uso de aulas virtuales, foros y tutoriales, para el aprendizaje y el desarrollo de competencias.

También publicar en el aula virtual, los mejores trabajos desarrollados por los estudiantes de la cátedra, con la finalidad de motivarlos a aportar más conocimientos y experiencias, contribuyendo a la formación de profesionales conocedores de las nuevas tendencias tecnológicas de la demanda y la oferta del mercado laboral, para el futuro desarrollo tecnológico de la sociedad y de nuestro país.

GLOSARIO

SQL

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en estas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre ella.

Servidor

Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes.

Bits: Los bits se utilizan para describir velocidades de transmisión.

Hardware: Corresponde a todas las partes tangibles de un sistema informático: sus componentes eléctricos, electrónicos, electromecánicos y mecánicos

Servidor: Es una computadora que, formando parte de una red, provee servicios a otras computadoras denominadas clientes

Software: Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático; comprende el conjunto de los componentes necesarios que hacen posible la realización de tareas específicas.

Web: Sistema de distribución de información basado en hipertexto accesible a través de Internet. Con un navegador web, un usuario visualiza páginas web que pueden contener texto, imágenes, videos u otros contenidos multimedia.

HTTP: (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.

GlassFish: es un servidor de aplicaciones de software libre

IDE: Integrated Development Environment ("entorno de desarrollo integrado", en español)

BIBLIOGRAFIA

Pérez, C. (2008). Administración y análisis de bases de datos (2° edición),

México D.F. Alfa Omega

Argueta, G.G.(2008). Guía de programación Oracle y Java Web. [Folleto].

El Salvador.

Argueta, G.G.(2008). Guía de programación Oracle y Java Web. [Folleto].

San Salvador.

Callejas, E. A. (2009). Guía de estudio de diseño de base de datos, [Folleto].

San Salvador.

Callejas, E. A. (2008). Guía de estudio Consulta de base de datos Oracle.

[Folleto]. San Salvador.

Callejas, E.A. (2010). Guía de estudio Manipulación de base de datos Oracle.

[Folleto]. San salvador.

Callejas E. A. (2010). Guía de estudio Manipulación de base de datos.

[Folleto]. San Salvador.

Sánchez F.V, Prudencio R. A y Ossio Marin S. W. (2008). Ejercicios de estudio Sentencias condicionales de Java, [Folleto]. San salvador.

Ajaxman, (2009) Para que sirve y que podemos hacer con Java, Recuperado de: <http://www.ajaxman.net/java-para-que-sirve-y-que-podemos-hacer-con-java.html>

GXtechnical. (2008). Java, requerimientos de software. Recuperado de: http://www.gxtechnical.com/gxdlsp/pub/genexus/java/docum/manuals/8.0/mjava_b2.htm

GXtechnical. (2008). Requerimientos Java. Recuperado de: <http://www.gxtechnical.com/gxdlsp/pub/genexus/internet/technicalpapers/..%5CDocum%5CManuals%5C7.5%5CReqJAVA.htm>

Isfesi, (2008). Como Instalar Oracle Database 11g para Windows. Recuperado de: <http://www.isfesi.es/juan/DAE/Instalar%20Oracle%20Database%2011g%20Standard%20Edition%20en%20Windows.pdf>

Java. (2008). ¿Qué es Java y porque lo necesito? Recuperado de: http://www.java.com/es/download/faq/whatis_java.xml

Java. (2008) ¿Que es Java? Recuperado de: http://java.com/es/download/whatis_java.jsp

Moodle. (2008). Instalación de Moodle. Recuperado de:

http://docs.moodle.org/all/es/Instalaci%C3%B3n_de_moodle

Ojedareinaldo. (2009). Requerimientos de Hardware para servidor de Moodle,

Recuperado de:

<http://ojedareinaldo.blogspot.com/2008/05/requerimientos-de-hardware-para-moodle.html>

Scribd. (2009). Requerimientos Java. Recuperado de:

<http://es.scribd.com/doc/36406625/149/Requerimientos-Java>

Slideshare. (2011). Instalación Oracle 11g, José guanuchi Recuperado de:

<http://www.slideshare.net/ppalberto/instalacion-oracle-11g-elaborado-por-jose-guanuchi>

Tecniscopio. (2009). Requerimientos del sistema NetBeans IDE 6.0.

Recuperado de: <http://tecniscopio.blogspot.com/2008/01/netbeans-ide-60-requerimientos-del.html>

Wikipedia. (2007). Oracle Database, característica de wikipedia, la enciclopedia libre. Recuperado de: http://es.wikipedia.org/wiki/Oracle_Database

XTEC. (2010). Instalación de Moodle. Recuperado de:

http://www.xtec.cat/~jqueralt/moodle/presentacio_lasalle.pdf

ANEXOS.

Consultas de base de datos ORACLE

a) Ejemplos de consultas simples con operadores aritméticos y relacionales.

1. Seleccionar los empleados con salario superior a 2000

```
select ename from emple where salario > 2000;
```

2. Seleccionar los empleados con salario superior a 2000 y con el trabajo de manager

```
select ename from emple where salario > 2000 and job like 'MANAGER';
```

3. Seleccionar el nº de departamento que tengan distinto nombre

```
select distinct name, deptno from emple;
```

4. Seleccionar todos los empleados del departamento nº 20, además debe de estar ordenado por el apellido y tienen que aparecer en la consulta el nº del empleado, el apellido, el oficio y el nº de departamento.

```
select empno, ename, job, deptno from emple where dept_no = 20 order  
by ename;
```

5. Empleados cuyo oficio sea analista, ordenado por el número del empleado.

```
select ename from emple where job like 'analista' order by empno;
```

6. Seleccionar las filas del departamento 10 cuyo oficio sea analista. La consulta debe de estar ordenada descendientemente por el apellido y también descendientemente por el nº empleado.

```
select deptno from emple where (job like 'analista' and order by empno  
desc, order by ename desc);
```

7. Seleccionar los nombres de los alumnos y su nota de la tabla notas_alumnos.

```
select nombre_alumno, notas from notas_alumnos;
```

b) Ejemplos de consultas simples con operadores aritméticos y relacionales.

8. Seleccionar los nombres de los alumnos y su nota de la tabla notas_alumnos siendo scott el propietario de la tabla.

```
select nombre_alumno, notas from scott.notas_alumnos;
```

9. Seleccionar la nota_media de los alumnos de la tabla notas_alumnos.

```
select nombre_alumno, (nota1+nota2+nota3)/3 from notas_alumnos;
```

10. Seleccionar los alumnos que de la nota1 hayan sacado un 7 y que la media de las 3 notas sea mayor que 6

```
select nota_alumno from notas_alumnos where nota1=7 and  
(nota1+nota2+nota3)/3>6;
```

11. Seleccionar los nombres de los alumnos y su nota siendo esta mayor o igual a 5 y su curso primero.

```
select nombre_alumno , notas from notas_alumnos where nota >=5  
and curso like 'primero';
```

12. Seleccionar los empleados que sean analistas

```
select ename from emple where job like 'analista';
```

13. Seleccionar los empleados cuyo apellido tenga una A en la 2ª posición.

```
select apellidos from emple where apellido like '_A%';
```

14. Seleccionar el apellido de los empleados cuyo oficio sea vendedor, analista o empleado

```
select apellido from emple where oficio like 'vendedor' or oficio like  
'analista' or 'oficio' like 'empleado';
```

15. Seleccionar el apellido, el salario y el nº de departamento de los empleados cuyo salario sea mayor o igual a 125000 en los departamentos 10 o 20.

```
select apellido, salario, dept_no from emple where salario > 125000  
and dept_no in (10,20);
```

16. Seleccionar la lista de empleados indicando para cada uno su apellido, oficio, fecha de alta y el salario con un aumento del 16%.

```
select apellido, oficio, fecha_al, salario* 1.16 from emple;
```

17. De cada departamento saber el nombre y la localidad.

```
select dnombre , loc from depart;
```

18. Seleccionar los departamentos agrupados por el nº departamento.

```
select * from depart order by depart_no asc;
```

19. Seleccionar los empleados ordenados alfabéticamente por el nombre y por el oficio.

```
select * from emple order by apellido asc, order by oficio asc;
```

20. Seleccionar los empleados que no tienen comisión.

```
select apellido from emple where comision = 0;
```

21. Seleccionar los empleados de apellido muñoz.

```
select apellido from emple where apellido like 'MUÑOZ';
```

22. Seleccionar los departamentos cuyo nombre acabe en 'on';

select dname from depart where dname like '%on';

23. Seleccionar los empleados cuyo nº de departamento no sea ni 30, ni 20 ni 40.

23.1) select apellidos from emple where dept_no<>30, dept_no<>20,
dept_no<>40;

23.2) select apellidos from emple where dept_no not in(30,20,40)

24. Seleccionar los departamentos cuya localidad no sea ni madrid ni bilbao.

select dname from depart where loc like not 'madrid',loc like not
'bilbao');

25. Seleccionar los empleados cuya fecha de alta este entre el 8/9/61 y el 27/2/82.

select apellido from emple where fecha_alt between '8/9/61' and
'27/2/82');

26. seleccionar los departamentos que terminen en n y tengan una o en su interior.

select dname from depart where dname like '%o%n' or dname like
'o%n';

27. Seleccionar los empleados que tengan una a en el nombre y una b en su población en la posición que sea.

```
select apellido from emple where apellido like '%a%' and poblacion  
like '%b%';
```

28. seleccionar los empleados cuyo salario sea mayor de 20000 y menor que 50000.

```
select apellidos from emple where salario>20000 and salario <  
50000;
```

Manipulación de base de datos ORACLE

a) Ejemplos de inserción de datos en tablas.

```
INTO      DEPART  VALUES  (10,'CONTABILIDAD','SEVILLA');  
INTO      DEPART  VALUES  (20,'INVESTIGACIÓN','MADRID');  
INTO      DEPART  VALUES  (30,'VENTAS','BARCELONA');  
INTO      DEPART  VALUES  (40,'PRODUCCIÓN','BILBAO');
```

```
INSERT INTO EMPL  VALUES (7369,'SÁNCHEZ','EMPLEADO',7902,'17/12/1980',  
104000,NULL,20);  
INSERT INTO EMPL  VALUES (7499,'ARROYO','VENDEDOR',7698,'20/02/1980',  
208000,39000,30);  
INSERT INTO EMPL  VALUES (7521,'SALA','VENDEDOR',7698,'22/02/1981',  
162500,65000,30);  
INSERT INTO EMPL  VALUES (7566,'JIMÉNEZ','DIRECTOR',7839,'02/04/1981',
```

```
386750,NULL,20);
```

```
insert into NOTAS_ALUMNOS VALUES ('Alcalde García, M. Luisa',5,5,5);
```

```
insert into NOTAS_ALUMNOS VALUES ('Benito Martín, Luis',7,6,8);
```

```
insert into NOTAS_ALUMNOS VALUES ('Casas Martínez, Manuel',7,5,5);
```

```
insert into NOTAS_ALUMNOS VALUES ('Corregidor Sánchez, Ana',6,9,8);
```

```
insert into NOTAS_ALUMNOS VALUES ('Díaz Sánchez, María',NULL,NULL,7);
```

```
INSERT INTO ASIGNATURAS VALUES (1,'Prog. Leng. Estr.');
```

```
INSERT INTO ASIGNATURAS VALUES (2,'Sist. Informáticos');
```

```
INSERT INTO ASIGNATURAS VALUES (3,'Análisis');
```

```
INSERT INTO ASIGNATURAS VALUES (4,'FOL');
```

```
INSERT INTO ALUMNOS VALUES
```

```
('12344345','Alcalde García, Elena', 'C/Las Matas, 24','Madrid','917766545');
```

```
INSERT INTO ALUMNOS VALUES
```

```
('4448242','Cerrato Vela, Luis', 'C/Mina 28 - 3A', 'Madrid','916566545');
```

```
INSERT INTO ALUMNOS VALUES
```

```
('56882942','Díaz Fernández, María', 'C/Luis Vives 25', 'Móstoles','915577545');
```

```
INSERT INTO NOTAS VALUES('12344345', 1,6);
```

```
INSERT INTO NOTAS VALUES('12344345', 2,5);
```

```
INSERT INTO NOTAS VALUES('12344345', 3,6);
```

```
INSERT INTO NOMBRES VALUES('PEDRO', 17);
```

```
INSERT INTO NOMBRES VALUES('JUAN', 17);
```

```
INSERT INTO NOMBRES VALUES('MARÍA', 16);
```

b) Ejemplos de eliminación de tablas.

```
DROP TABLE DEPART cascade constraints;
```

```
DROP TABLE EMPLE cascade constraints;
```

```
Drop table notas_alumnos cascade constraints;
```

```
Drop table LIBRERIA cascade constraints;
```

```
DROP TABLE ALUMNOS cascade constraints;
```

```
DROP TABLE ASIGNATURAS cascade constraints;
```

```
DROP TABLE NOTAS cascade constraints;
```

```
DROP TABLE NOMBRES cascade constraints;
```

DROP TABLE MISTEXTOS cascade constraints;
DROP TABLE LIBROS cascade constraints;
DROP TABLE NACIMIENTOS cascade constraints;
DROP TABLE PARALEER cascade constraints;

DROP TABLE LEIDOS cascade constraints;

DROP TABLE ALUM cascade constraints;

DROP TABLE NUEVOS cascade constraints;

DROP TABLE ANTIGUOS cascade constraints;

DROP TABLE PERSONAL cascade constraints;

DROP TABLE PROFESORES cascade constraints;

DROP TABLE CENTROS cascade constraints;

DROP TABLE EMPLE30 cascade constraints;

DROP TABLE COCHES cascade constraints;

c) Ejemplos de Inserción, eliminación y actualización mediante consultas.

1. Dadas las tablas ALUM y NUEVOS, insertar en la tabla ALUM los nuevos alumnos.

```
insert into alum (select * from nuevos minus select * from alum);
```

2. Borrar de la tabla ALUM los ANTIGUOS alumnos.

```
delete from alum where nombre in (select nombre from antiguos);
```

3. Insertar a un empleado de apellido 'SAAVEDRA' con número 2000. La fecha de alta será la actual, el SALARIO será el mismo salario de 'SALA' mas el 20 por 100 y el resto de datos serán los mismos que los datos de 'SALA'.

```
insert into emple (select 2000,'SAAVEDRA',oficio,dir,SYSDATE,salario +  
salario*0.2,comision,dept_no from emple where apellido like 'SALA');
```

4. Modificar el número de departamentos de 'SAAVEDRA'. El nuevo departamento será el departamento donde hay más empleados cuyo oficio sea 'EMPLEADO'

```
update emple set dept_no = (select dept_no from emple where oficio like  
'EMPLEADO' group by dept_no having count (*) = (select max(count(*))  
from emple where oficio like 'EMPLEADO' group by dept_no)) where  
apellido like 'SAAVEDRA';
```

5. Borrar todos los departamentos de la tabla DEPART para los cuales no existan empleados en emple.

```
delete from depart where dept_no in (select dept_no from depart minus
select dept_no from emple);
```

6. Modificar el número de plazas con un valor igual a la mitad en aquellos centros con menos de dos profesores.

```
update centros set num_plazas = num_plazas/2 where cod_centro in
(select cod_centro from profesores group by cod_centro having
count(*)<2);
```

7. Eliminar los centros que no tengan personal.

```
delete centros where cod_centro in (select cod_centro from centros minus
select cod_centro from personal);
```

8. Añadir un nuevo profesor en el centro o en los centros cuyo número de administrativos sea 2 en la especialidad de 'IDIOMA', con DNI 8790055 y de nombre 'Clara Salas'.

```
insert into profesores select distinct cod_centro,
8790055,'Salas,Clara','IDIOMA' from personal where cod_centro in (select
cod_centro from personal where funcion = 'ADMINISTRATIVO' group by
cod_centro having count(*) = 1);
```

9. Borrar al personal que esté en centros de menos de 300 plazas y con menos de dos profesores.

```
delete personal where cod_centro in (select cod_centro from centros
where num_plazas<300) and cod_centro in (select cod_centro from
profesores group by cod_centro having count(*)<2);
```

10. Borrar a los profesores que estén en la tabla PROFESORES y que no estén en la tabla PERSONAL.

```
delete profesores where dni not in (select dni from personal);
```

11. Dar de alta un nuevo artículo de 'Primera' categoría para los fabricantes de 'Francia' y abastecer con 5 unidades de ese artículo a todas las tiendas y en la fecha de hoy.

```
insert into articulos select 'Yogur Fresa', cod_fabricante, 4, 'Primera',
120,100,190 from fabricantes where pais = 'FRANCIA'; select nif,
'Yogur Fresa', cod_fabricante, 4, 'Primera', sysdate, 5 from tiendas,
fabricantes where pais like 'FRANCIA';
```

12. Insertar un pedido de 20 unidades en la tienda '111-A' con el artículo que mayor número de ventas haya realizado.

```
insert into pedidos select distinct '111-A', articulo, cod_fabricante,
peso, categoria, sysdate, 20 from ventas where (articulo,
```

```
cod_fabricante, peso, categoria) in (select articulo, cod_fabricante,
peso, categoria from ventas group by articulo cod_fabricante, peso,
categoria having count (*) = (select max(count(*)) from ventas group
by articulo, cod_fabricante, peso, categoria));
```

13. Dar de alta una tienda en la provincia de 'MADRID' y abastecerla con 20 unidades de cada uno de los artículos existentes.

```
insert into tiendas values ('1010-C', 'La Cesta', 'C/Juan Mazo 30',
'Alcala','MADRID',28809);
```

```
insert into pedidos select '1010-C', articulo, cod_fabricante, peso,
categoria, sysdate, 20 from articulos;
```

14. Dar de alta dos tiendas en la provincia de 'SEVILLA' y abastecerlas con 30 unidades de artículos de la marca 'GALLO'

```
insert into tiendas values('4501-B','La Económica','C\Sevillanas
130','Dos hermanas','SEVILLA',44003);
```

```
insert into tiendas values('4501-B','Cometibles Peter','C\Sevillanas
130','Dos hermanas','SEVILLA',44009);
```

```
insert into pedidos select nif, articulo, a.cod_fabricante, peso,
categoria, sysdate, 30 from tiendas, artículos a, fabricantes f where
```

```
provincia='SEVILLA' and f.nombre='GALLO' and  
a.cod_fabricante=f.cod_fabricante;
```

15. Realizar una venta para todas las tiendas de 'TOLEDO' de 10 unidades en los artículos de 'Primera' categoría.

```
insert into ventas select nif, articulo, cod_fabricante, peso, categoria,  
sysdate, 10 from tiendas, artículos where provincia='TOLEDO' AND  
categoria='Primera';
```

16. Para aquellos artículos de los que se hayan vendido más de 30 unidades, realizar un pedido de 10 unidades para la tienda con NIF '5555-B' con la fecha actual.

```
insert into pedidos select distinct '5555-B', articulo, cod_fabricante,  
peso, categoria, sysdate, 10 from ventas where  
(articulo,cod_fabricante,peso,categoria) in (select  
articulo,cod_fabricante,peso,categoria from ventas group by articulo,  
cod_fabricante,peso,categoria having sum(unidades_vendidas)>30);
```

17. Cambiar los datos de la tienda con NIF '1111-A' igualándolos a los de la tienda NIF '2222-A'.

```
update tiendas  
set(nombre,dirección,población,provincia,codpostal)=(select nombre,di
```

```
rección,población,provincia,codpostal    from    tiendas    where  
pais='ITALIA');
```

18. Cambiar todos los artículos de 'Primera' categoría a 'Segunda' categoría del país 'ITALIA'.

```
UPDATE ARTICULOS SET CATEGORIA='Segunda'
```

```
WHERE CATEGORIA='Primera'
```

```
AND COD_FABRICANTE = (SELECT COD_FABRICANTE FROM  
FABRICANTES WHERE PAIS='ITALIA');
```

19. Modificar aquellos pedidos en los que la cantidad pedida sea superior a las existencias del artículo, asignando el 20% de las existencias a la cantidad que se ha pedido.

```
UPDATE PEDIDOS P
```

```
SET UNIDADES_PEDIDAS=
```

```
(SELECT EXISTENCIAS*0.2 FROM ARTICULOS WHERE
```

```
ARTICULO=P. ARTICULO AND
```

```
COD_FABRICANTE=P. COD_FABRICANTE AND
```

```
PESO=P. PESO AND
```

```
CATEGORIA=P. CATEGORIA )
```

```
WHERE UNIDADES_PEDIDAS >  
  
(SELECT EXISTENCIAS FROM ARTICULOS WHERE  
  
ARTICULO=P. ARTICULO AND  
  
COD_FABRICANTE=P. COD_FABRICANTE AND  
  
PESO=P. PESO AND  
  
CATEGORIA=P. CATEGORIA);
```

20. Eliminar aquellas tiendas que no han realizado ventas.

```
DELETE TIENDAS  
  
WHERE NIF NOT IN (SELECT DISTINCT NIF FROM VENTAS);
```

21. Eliminar los artículos que no hayan tenido ni compras ni ventas.

```
DELETE ARTICULOS  
  
WHERE (ARTICULO, COD_FABRICANTE, PESO, CATEGORIA)  
  
NOT IN (SELECT DISTINCT ARTICULO, COD_FABRICANTE,  
PESO,  
  
CATEGORIA FROM VENTAS)  
  
AND (ARTICULO, COD_FABRICANTE, PESO, CATEGORIA)
```

```
NOT IN (SELECT DISTINCT ARTICULO, COD_FABRICANTE,  
PESO,  
  
CATEGORIA FROM PEDIDOS);
```

22. Borrar los pedidos de 'Primera' categoría cuyo país de procedencia sea 'BELGICA'.

```
DELETE PEDIDOS  
  
WHERE (ARTICULO, COD_FABRICANTE, PESO, CATEGORIA)  
  
IN  
  
(SELECT ARTICULO, COD_FABRICANTE, PESO, CATEGORIA  
  
FROM ARTICULOS WHERE COD_FABRICANTE=  
  
(SELECT COD_FABRICANTE FROM FABRICANTES WHERE PAIS  
='BELGICA'))  
  
AND CATEGORIA='Primera';
```

23. Borrar los pedidos que no tengan tienda.

```
delete pedidos where nif not in(select nif from tiendas);
```

24. Restar uno a las unidades de los últimos pedidos de la tienda con NIF '5555-B'.

```
UPDATE PEDIDOS
```

```
SET UNIDADES_PEDIDAS=UNIDADES_PEDIDAS-1
```

```
WHERE NIF='5555-B' AND
```

```
FECHA_PEDIDO=
```

```
(SELECT MAX(FECHA_PEDIDO) FROM PEDIDOS WHERE  
NIF='5555-B');
```

d) Ejemplos de creación de tablas y modificación de las tablas.

1º). Crear la tabla fabricantes con las siguientes condiciones:

- La clave primaria es cod_fabricante
- Las columnas nombre y país han de almacenarse en mayúsculas

```
CREATE TABLE FABRICANTES(
```

```
COD_FABRICANTE NUMBER(3) CONSTRAINT PK_FA PRIMARY KEY,
```

```
NOMBRE VARCHAR2(15)
```

```
CONSTRAINT CK_NO CHECK(NOMBRE=UPPER(NOMBRE)),
```

```
PAIS VARCHAR2(15)
```

```
CONSTRAINT CK_PA CHECK(PAIS=UPPER(PAIS))
```

);

1º.1.) Crear la tabla artículos con las siguientes condiciones:

- La clave primaria está formada por las columnas: articulo, cod_fabricante, peso y categoría.
- cod_fabricante es clave ajena que referencia a la tabla fabricantes.
- precio_venta, precio_costo han de ser >0.
- categoría ha de ser 'primera','segunda ' o 'tercera'.

```
CREATE TABLE ARTICULOS(  
  
    ARTICULO VARCHAR2(20)NOT NULL,  
  
    COD_FABRICANTE NUMBER(3) NOT NULL,  
  
    PESO NUMBER(3) NOT NULL  
  
    CONSTRAINT CK1_AR CHECK (PESO>0),  
  
    CATEGORIA VARCHAR2(10) NOT NULL,  
  
    PRECIO_VENTA NUMBER (4)  
  
    CONSTRAINT CK2_AR CHECK (PRECIO_VENTA>0),  
  
    PRECIO_COSTO NUMBER (4)  
  
    CONSTRAINT CK3_AR CHECK (PRECIO_COSTO>0),  
  
    EXISTENCIAS NUMBER (5),
```

```
CONSTRAINT PK_ART PRIMARY KEY
```

```
(ARTICULO, COD_FABRICANTE, PESO, CATEGORIA),
```

```
CONSTRAINT FK_ARFA FOREIGN KEY (COD_FABRICANTE)
```

```
REFERENCES FABRICANTES,
```

```
CONSTRAINT CK_CAT CHECK(CATEGORIA IN('Primera','Segunda',  
'Tercera'))
```

```
);
```

1º.3.) Crear la tabla tiendas con las siguientes condiciones:

- la clave primaria es NIF
- provincia ha de almacenarse en mayúsculas.

```
CREATE TABLE TIENDAS(
```

```
NIF VARCHAR2(10) NOT NULL CONSTRAINT PK_TI PRIMARY KEY,
```

```
NOMBRE VARCHAR2(20),
```

```
DIRECCIÓN VARCHAR2(20),
```

```
POBLACIÓN VARCHAR2(20),
```

```
PROVINCIA VARCHAR2(20)
```

```
CONSTRAINT CK_PRO CHECK(PROVINCIA=UPPER(PROVINCIA)),  
  
CODPOSTAL NUMBER(5)  
  
);
```

1º.4.) Crear la tabla pedidos con las siguientes condiciones:

- La clave primaria está formada por las columnas: NIF, articulo,cod_fabricante,peso,categoria y fecha de pedido.
- cod_febricante es clave ajena que referencia a la tabla fabricantes.
- unidades pedidas ha de ser: >0
- categoría ha de ser 'primera','segunda ' o 'tercera'.
- las columnas artículo, cod_fabricante, peso y categoría son clave ajena y referencia a la tabla artículos. Realizar un borrado en cascada.
- Nif es clave ajena y referencia a la tabla tiendas.

```
CREATE TABLE PEDIDOS(  
  
NIF VARCHAR2(10) NOT NULL  
  
CONSTRAINT FK_PETI REFERENCES TIENDAS,  
  
ARTICULO VARCHAR2(20) NOT NULL,  
  
COD_FABRICANTE NUMBER(3) NOT NULL  
  
CONSTRAINT FK_PEFA REFERENCES FABRICANTES,
```

```

PESO NUMBER(3) NOT NULL,

CATEGORIA VARCHAR2(10) NOT NULL,

FECHA_PEDIDO DATE NOT NULL,

UNIDADES_PEDIDAS NUMBER(4)

CONSTRAINT CK_PEUP CHECK (UNIDADES_PEDIDAS>0),

CONSTRAINT PK_PED PRIMARY KEY

(NIF,ARTICULO,      COD_FABRICANTE,      PESO,      CATEGORIA,
FECHA_PEDIDO),

CONSTRAINT          CK_CATPE          CHECK(CATEGORIA
IN('Primera','Segunda','Tercera')),

CONSTRAINT FK_PEAR FOREIGN KEY

(ARTICULO, COD_FABRICANTE, PESO, CATEGORIA) REFERENCES
ARTICULOS

ON DELETE CASCADE

);

```

1º.4.) Crear la tabla ventas con las siguientes condiciones:

- La clave primaria está formada por la por las columnas: nif, articulo,cod_fabricante,peso,categoria y fecha venta.
- cod fabricante es clave ajena que referencia a la tabla fabricantes.
- unidades_vendidas ha de ser >0
- categoría ha de ser 'primera','segunda ' o 'tercera'.
- Las columnas artículo, cod_fabricante, peso, y categoría son clave ajena y referencia de la tabla artículos. Realizar un borrado en cascada.
- NIF es clave ajena y referencia a la tabla tiendas

```
CREATE TABLE VENTAS(

NIF VARCHAR2(10) NOT NULL

CONSTRAINT FK_VETI REFERENCES TIENDAS,

ARTICULO VARCHAR2(20) NOT NULL,

COD_FABRICANTE NUMBER(3) NOT NULL

CONSTRAINT FK_VEFA REFERENCES FABRICANTES,

PESO NUMBER(3) NOT NULL,

CATEGORIA VARCHAR2(10) NOT NULL

CONSTRAINT CK_CATVE

CHECK(CATEGORIA IN('Primera','Segunda','Tercera'))),
```

FECHA_VENTA DATE NOT NULL,

UNIDADES_VENDIDAS NUMBER(4)

CONSTRAINT CK_VEUV CHECK (UNIDADES_VENDIDAS>0),

CONSTRAINT PK_VEN PRIMARY KEY

(NIF,ARTICULO, COD_FABRICANTE, PESO, CATEGORIA,
FECHA_VENTA),

CONSTRAINT FK_VEAR FOREIGN KEY

(ARTICULO, COD_FABRICANTE, PESO, CATEGORIA) REFERENCES
ARTICULOS

ON DELETE CASCADE

);

- 2º). Añadir una restricción a la tabla tiendas para que el nombre de la tienda sea de tipo título

ALTER TABLE TIENDAS ADD CONSTRAINT

NOMBRETITU CHECK (NOMBRE=INITCAP(NOMBRE));

- 3º). Visualizar las constraint definidas para las tablas anteriores.

```

SELECT    TABLE_NAME,    CONSTRAINT_NAME    FROM
USER_CONSTRAINTS

WHERE      TABLE_NAME      IN('ARTICULOS','TIENDAS',
'FABRICANTES', 'PEDIDOS', 'VENTAS');

```

- 4º). Modificar las columnas de las tablas pedidos y ventas para que las unidades vendidas y las unidades pedidas puedan almacenar cantidades numéricas de 6 dígitos.

```

ALTER TABLE PEDIDOS MODIFY (UNIDADES_PEDIDAS
NUMBER(6));

ALTER TABLE VENTAS MODIFY (UNIDADES_VENDIDAS
NUMBER(6));

```

- 5º). Impedir que se den más de alta mas tiendas en la provincia de toledo.

```

ALTER TABLE TIENDAS ADD CONSTRAINT CK_TIENTO

CHECK (PROVINCIA !='TOLEDO');

```

- 6º). Añadir a las tablas pedidos y ventas una nueva columna para que almacenen el pvp dep artículo.

```

ALTER TABLE PEDIDOS ADD(PVP NUMBER(4));

ALTER TABLE VENTAS ADD(PVP NUMBER(4));

```

TABLAS PERSONAL, PROFESORES y CENTROS:

- 9º). Añadir a la tabla profesores una columna llamada cod_asig con dos posiciones numéricas.

```
ALTER TABLE PROFESORES ADD(COD_ASIG NUMBER(2));
```

- 10º). Crear la tabla tasig con las siguientes columnas: cod_asig numerico, 2 posiciones y nom_asig cadena de 20 caracteres.

```
CREATE TABLE TASIG (  
  
    COD_ASIG NUMBER(2),  
  
    NOM_ASIG VARCHAR2(20) );
```

- 11º). Añadir la restricción de clave primaria a la columna cod_asig de la tabla tasig.

```
ALTER TABLE TASIG  
  
ADD CONSTRAINT PK_TASIG PRIMARY KEY (COD_ASIG);
```

- 12º). Añadir la restricción de la clave ajena a la columna cod_asig de la tabla profesores.

```
ALTER TABLE PROFESORES  
  
ADD CONSTRAINT FK_TASIG FOREIGN KEY (COD_ASIG)  
  
REFERENCES TASIG;
```

- 13º). Visualizar los nombres de las constraint y las columnas afectadas de las tablas tasig y profesores.

```
SELECT TABLE_NAME, CONSTRAINT_NAME, COLUMN_NAME  
  
FROM    USER_CONS_COLUMNS    WHERE    TABLE_NAME  
  
IN('TASIG','PROFESORES');
```

- 14º). Cambiar los nombres de las tablas profesores y llamar las profes

```
RENAME PROFESORES TO PROFES;
```

- 15º). Borrar la tabla tasig

```
DROP TABLE TASIG CASCADE CONSTRAINTS;
```

- 16º). Devolver la tabla profesores a su situación inicial

```
CREATE TABLE PROFESORES AS  
  
SELECT COD_CENTRO, DNI, APELLIDOS, ESPECIALIDAD  
  
FROM PROFES;
```

Diseño de base de datos

1. La ferretería los tres ases necesita que le diseñen una base de datos que le permita mantener la información de todas las ventas, que se realizan en la ferretería. Se necesita, que la base de datos controle la información de productos vendidos y su disponibilidad del inventario para ventas, además le

interesa controlar quienes de sus empleados, son más eficientes en las ventas de sus productos y cuáles son sus clientes,

2. Diseñe la base de datos relacional para un sistema donde se desea almacenar la información de una compañía aérea. La compañía tiene tres recursos principales: Aviones, pilotos, tripulación. De los pilotos se almacena su código, nombre y horas de vuelo. De los miembros de la tripulación solo se tendrá el código y el nombre. Pilotos y tripulación tienen una base a la que regresan después de cada jornada. Un vuelo va desde un origen a un destino a una hora específica de salida y de llegada, así como la fecha del vuelo. Cada avión tiene un código, es de un tipo (Boing, Airbus, entre otros), además tiene una base donde es sometido a mantenimiento.

3. Diseñe una base de datos relacional para una biblioteca donde se manejan fichas de autores y libros. En la ficha de cada autor se tiene el código de autor y el nombre. De cada libro se guarda el código, título, ISBN, editorial y número de página. Un autor puede escribir varios libros, y un libro puede ser escrito por varios autores. Un libro está formado por ejemplares. Cada ejemplar tiene un código y una localización. Un libro tiene muchos ejemplares y un ejemplar pertenece sólo a un libro.

Los usuarios de la biblioteca también disponen de ficha en la biblioteca y prestan ejemplares de ella. De cada usuario se guarda el código, nombre, dirección y teléfono.

Los ejemplares son prestados a los usuarios. Un usuario puede tomar prestados varios ejemplares, y un ejemplar puede ser prestado a varios usuarios. De cada préstamo interesa guardar la fecha de préstamo y la fecha de devolución”.

4. La empresa la “La Ideosa” compra productos por internet a sus proveedores locales y extranjeros, los cuales son ofrecidos en almacenes locales, a los clientes que los compran por detalle y mayoreo. Estos pagan al crédito y al contado con cheque, efectivo y tarjeta de crédito. Algunos procesos son: El cliente llega a la empresa para comprar el producto por mayor o por detalle, luego el producto es facturado y cliente paga al por cheque, tarjeta de crédito o en efectivo. Cuando son ventas muy grandes, La empresa hace un envío de la mercadería a la dirección del cliente.
5. Realizar un diseño de Base de Datos, para una aplicación de INSCRIPCION DE MATERIAS DE LA UNIVERSIDAD TECNOLÓGICA DE EL SALVADOR. El alumno inscribirá sus materias en línea a través del portal de la Universidad, tomando en cuenta que este se encuentre solvente en cuanto a pagos y en la biblioteca; no podrá inscribir materias que no se han cursado sus prerrequisitos y tampoco puede inscribir materias en 4ª. Matrícula. El alumno por ningún motivo podrá inscribir materias que no pertenezcan a su pensum.

6. MARN decide crear un sistema de información sobre parques naturales gestionados por cada comunidad autónoma, (CA) con los siguientes requerimientos. Una CA puede tener varios parques, en toda la CA existe un único organismo responsable de los parques, un parque puede estar compartido por más de una comunidad. Un parque natural se identifica por un nombre y la fecha en que fue declarado parque natural, y está compuesto por varias áreas identificadas por un nombre y una extensión en kilómetros, En cada área residen especies que pueden ser de tres tipos: vegetales, animales y minerales, Cada especie tiene una denominación científica y una vulgar. De las especies vegetales se quiere saber si tiene flora y en qué mes se produce. De los animales se quiere saber su alimentación y su periodo de celo. De los minerales saber si son rocas o cristales. Además que especies sirven de alimento a otra especie. Cada parque tiene su personal en dedicación exclusiva. De este personal se guarda DUI, numero de ISSS, nombre, dirección, teléfono y sueldo. De un visitante se desea guardar el DUI, nombre, dirección y profesión. Además puede alojarse en los alojamientos que tiene el parque. Los alojamientos tienen una capacidad limitada y una determinada categoría. Los alojamientos organizan excursiones al parque en vehículo o a pie unos determinados días de la semana y a horas determinadas. Para ir a estas excursiones hay que ser visitante del parque.

7. Sea una empresa dedicada al alquiler de CD-ROMs de audio. Dicha empresa tiene un local de atención al público donde están expuestas las carátulas de los CDs más demandados y las últimas novedades, aunque también existen listados en papel de todos los títulos que se podrían alquilar. Cuando un cliente solicita en alquiler un título, se comprueba si hay ejemplares disponibles y si el cliente no tiene problemas por ejemplares no devueltos, quedando constancia de la fecha de alquiler y la fecha máxima de entrega; de forma que cuando el cliente devuelva el ejemplar se podrá comprobar si se le tiene que imponer una sanción. Cada cliente puede solicitar una relación de los CDs que ha alquilado previamente.

Cada ejemplar de cada título debe quedar plenamente identificado (incluyendo la información necesaria para su rápida localización física).

Tablas(Cliente-Titulo-Sancion- CD-Alquiler-Detalle Alquiler)

8. Un centro de enseñanza dedicado a impartir cursos especializados de Informática desea automatizar la gestión de dicho cursos y para ello solicita una BD que permita la recopilación y posterior tratamiento de dicha información. Básicamente se busca tener la información de los cursos y de los recursos disponibles para impartirlos como los horarios, profesores, material didáctico, recursos multimedia y la disponibilidad de aulas para ser impartidos. Proveyendo luego la capacidad del aula para definir la cantidad de alumnos a inscribir en cada curso.

9. La empresa “Proyectos, S.A.” desea diseñar una BD para poder controlar los proyectos en los que sus empleados están asignados o trabajando, además de saber que puestos ocupan y a qué departamento pertenece dentro de la empresa y que departamentos ha solicitado el proyecto y el avance de cada proyecto en curso. Se necesita tener a la mano los datos generales de los Empleados (dui, nombre, teléfono, dirección, etc.), Proyectos (nombre, presupuesto, avance, etc.) e información general de los departamentos para las cuales se trabaja.
10. En una empresa dedicada a la fabricación de productos para el hogar se necesita elaborar una base de datos que almacene información sobre la producción diaria de cojines de diferentes tipos. Esta base de datos debe permitir almacenar:
- a. Información de los proveedores
 - b. Volumen de producción diaria
 - c. Características de los cojines que se fabrican
 - d. Información sobre las personas que producen los cojines
 - e. Costo por materia prima por proveedor
 - f. Información sobre la materia prima utilizada
 - g. Clasificaciones de cojines

h. Costo de almacenamiento de productos fabricados

i. Datos de la empresa

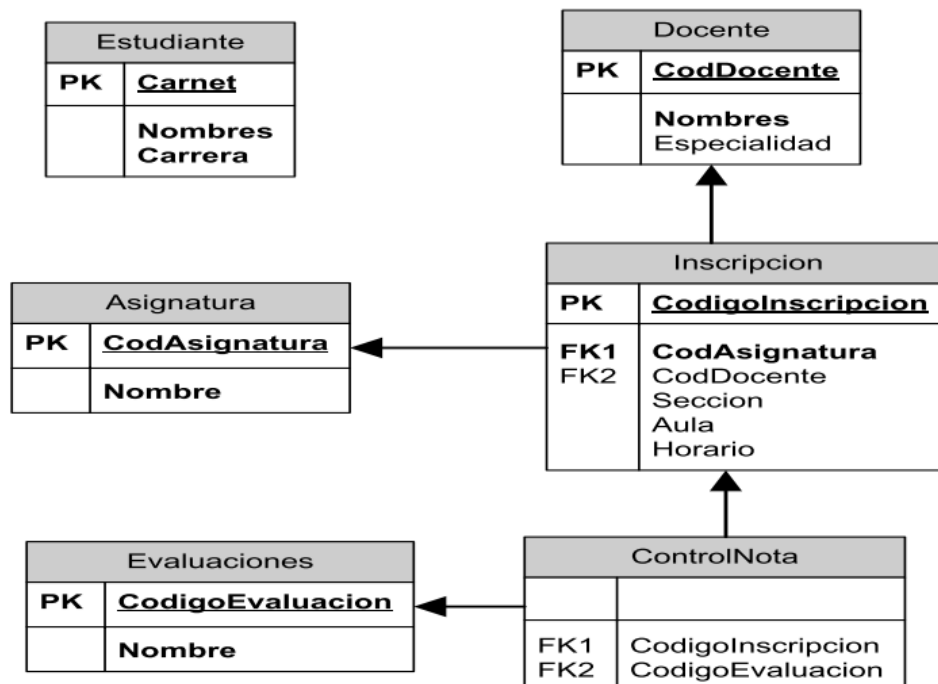
Manipulación de base de datos

Ejercicio # 1

1. Crear una base de datos llamada de la siguiente manera:

TBS_NombreApellido, es decir si su nombre es Juan Perez, el tablespace se llamara JuanPerez y crearla en la ubicación c:\bd; el usuario será own_NombreApellido

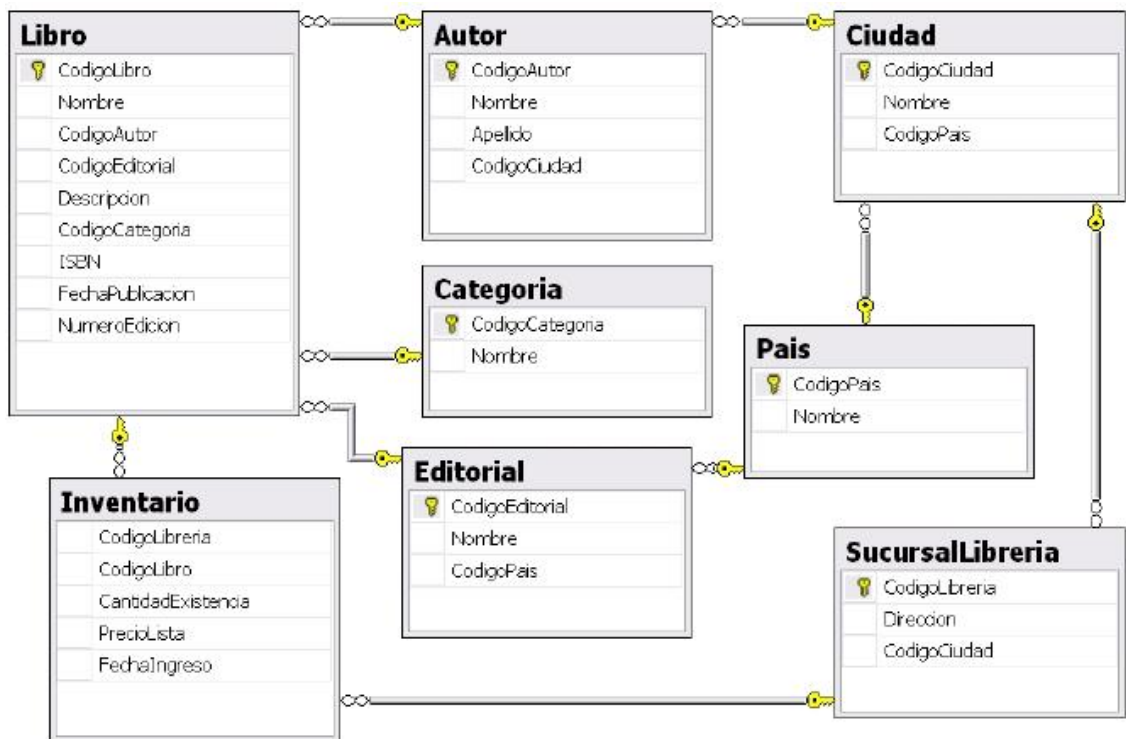
2. Crear la siguiente estructura de las tablas siguientes



3. Agregar una columna en la tabla inscripción llamada carnet de tipo char(10)
4. Crear un constraint en el campo carnet llamado fk_EstudianteInscripcion que haga referencia a la tabla Estudiante
5. Agregar un campo en la tabla estudiante llamado Fecha Ingreso y que tenga como valor por defecto la fecha del sistema operativo
6. Agregar un campo llamado nota de tipo number(8,2) en la tabla controlnota con una restricción mayor que 0 y menor que 10
7. Eliminar el constraint de la tabla docente e inscripción
8. Eliminar el campo horario de la tabla inscripción
9. Borrar la tabla Inscripción
10. Agregar una llave primaria en la tabla controlnota
11. Modificar el campo especialidad de tipo char en la tabla docente
12. Eliminar el constraint de la tabla estudiante
13. Crear las secuencias para cada tabla
14. Crear los trigger que disparen la generación del campo de llave primaria a través de la secuencia
15. Insertar registros en las diferentes tables del esquema de la base de datos
NombreApellido

Ejercicio # 2

La Librería “El Gran Lector” está desarrollando un sistema informático que le permita mantener el inventario de sus libros disponibles para la venta. Para ello, se muestra a continuación parte de la base de datos que utilizará



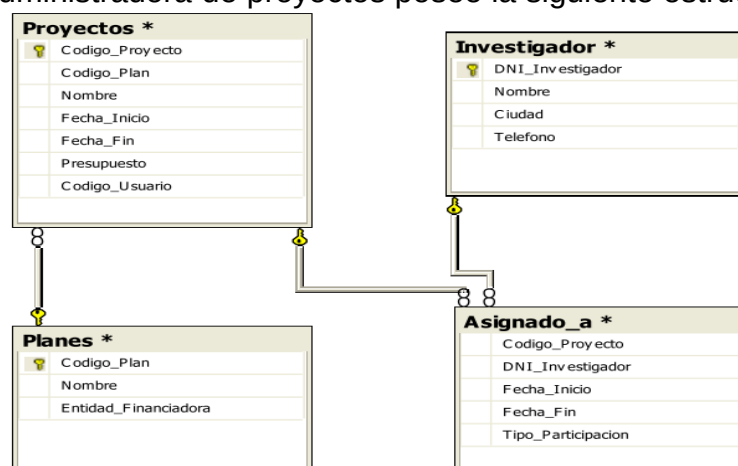
Se pide:

1. Crear la base de datos con el usuario own_libreria en tbs_Libreria
2. Agregar un datafile a tbs_libreria
3. Insertar registros a los campos de la base de datos libreria

4. Colocar la del sistema operativo fecha de ingreso de inventario con valor por defecto de la fecha
5. El precio de lista de inventario agregar la restricción de ser mayor a cero
6. Eliminar la restricción de unión entre la tabla inventario y sucursal
7. Crear las secuencias y triggers que generen el código de cada una de las tablas
8. Insertar registros en cada tabla
9. Crear un usuario llamdo usr_libreria que tenga los permisos de consultar e insertar en las tablas inventario y libro
10. Crear un usuario llamado man_libreria que tenga los permisos de actualizar y eliminar libros de las tables inventario y libro

Ejercicio # 3

La oficina administradora de proyectos posee la siguiente estructura de base de datos

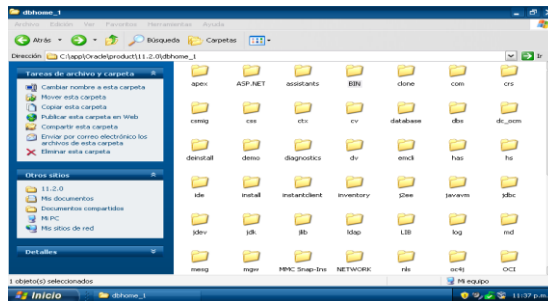


Se pide

1. Definir la bd con tres data file
2. Crear los perfiles de usuario para consultar las tablas, usuario para manipular las tablas, usuario propietario de la base de datos de proyectos que pueda crear objetos en la base de datos
3. Establecer restricciones de integridad, valores por defecto, restricciones de valores esperados
4. crear las secuencias y triggers para las tablas
5. Insertar registros en cada una de las tablas
6. Eliminar restricciones sobre campos
7. Modificar tipos de datos
8. Eliminar constraint
9. Eliminar valores por defectos
10. Eliminar los objetos de la base de datos

Configuración de Listener

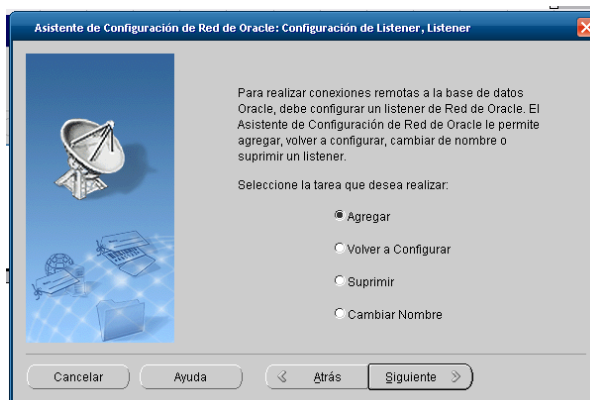
Para añadir un listener abriremos el Explorador de Windows, accederemos a la carpeta de instalación de Oracle:



Pulsaremos con el botón del ratón sobre el fichero "netca.bat" y seleccionaremos "Ejecutar"



Se iniciará el Asistente de Configuración de Red de Oracle, marcaremos la opción "Configuración del Listener" y pulsaremos "Siguiente":



Elegir la opción "Agregar" y pulsar "Siguiente":

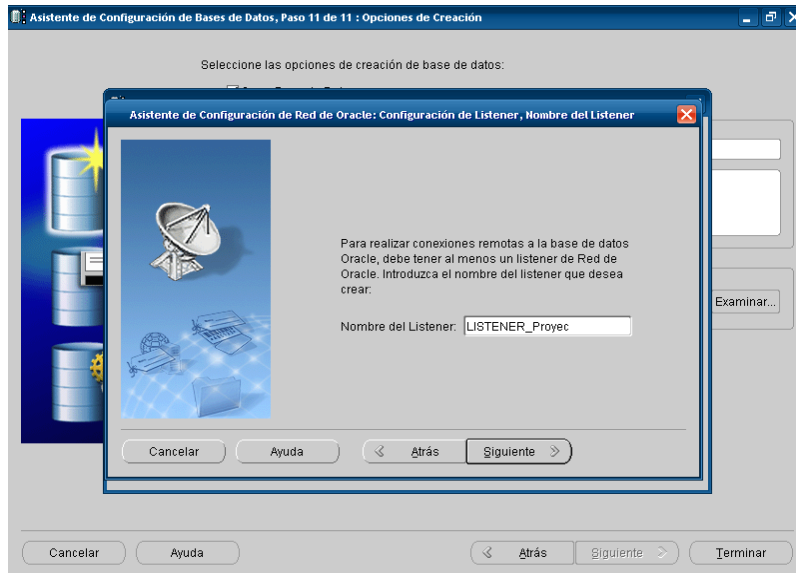


Figura 3.3.1.2.4

En "Nombre del Listener" introducir un nombre que lo identifique (si se tienen varios), por ejemplo "LISTENER_EJEMPLO"



Figura 3.3.1.2.5

Indicar el protocolo (o protocolos) que se considere que podrá aceptar este Listener, en este caso TCP y pulsar siguiente.



Figura 3.3.1.2.6

Indicar el puerto de Oracle Database, si no se ha cambiado manualmente, el de defecto es 1521, por lo que se marca "Usar el número de puerto estándar 1521" y presionar siguiente.

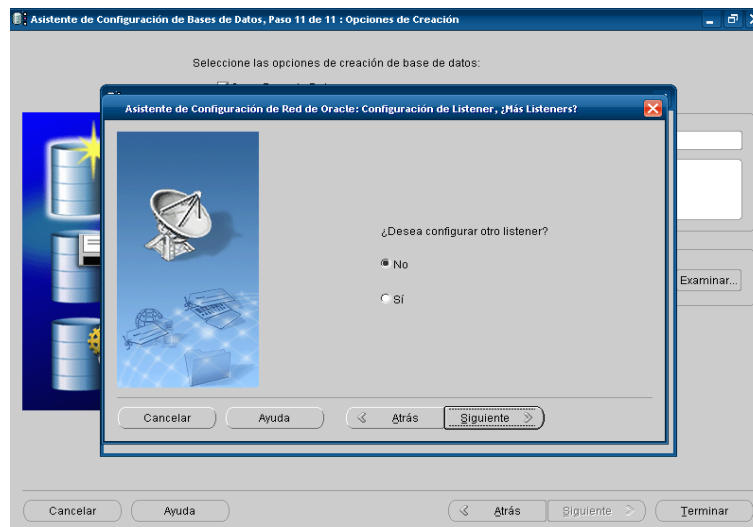


Figura 3.3.1.2.7

Marcar "No" en la siguiente ventana que nos permite crear otro Listener y pulsar siguiente.

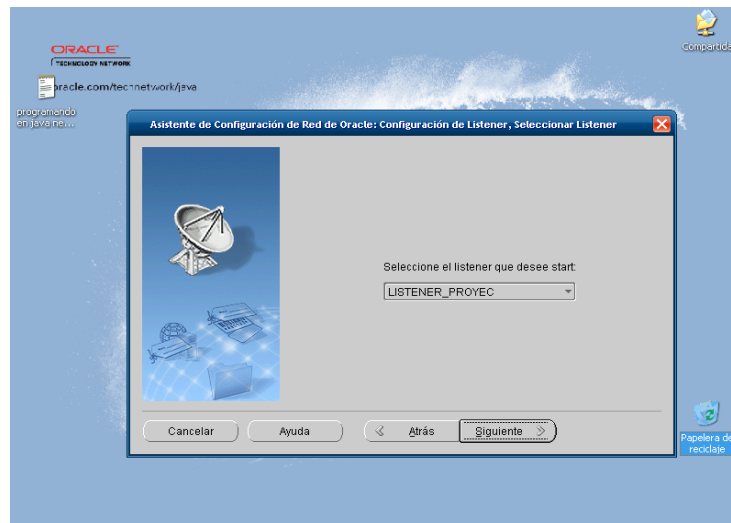


Figura 3.3.1.2.8

Seleccione el Listener que desea levantar y hacer clic en siguiente.

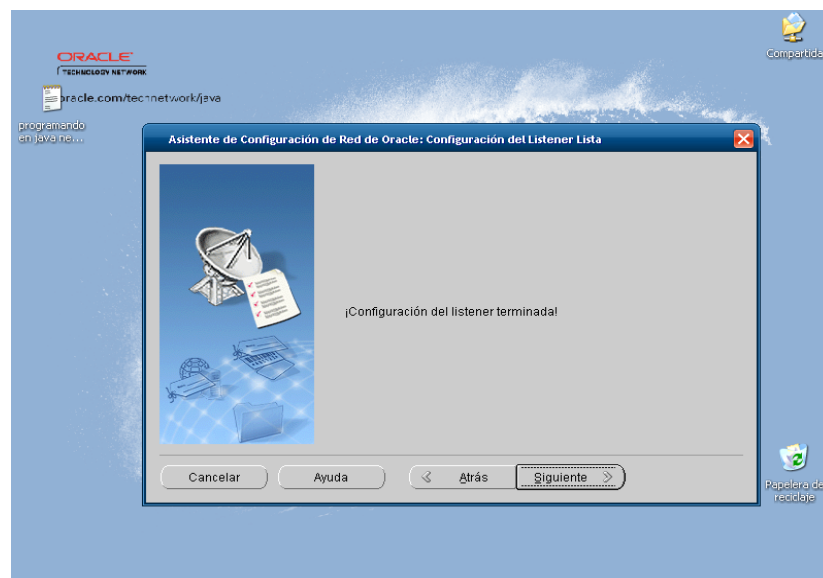


Figura 3.3.1.2.9

Presionar sobre el botón Siguiente, para finalizar la configuración.

Uso de sentencias condicionales de Java.

En los siguientes ejercicios, se utilizan las sentencias condicionales if-Else, además de sentencias aritméticas como % (modulo), comparativas (<,>==,etc) y también se hace uso de contadores y centinelas.

1. Dado un número entero y positivo que se introduce por teclado, determinar si es par o impar

EXPLICACION DEL PROGRAMA: Este programa lee un numero introducido por teclado y verifica si el numero es par o impar, verificando si el modulo del numero entre 2 es 0 o no.

```
package ejercicio_01;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Ejercicio_01 {

    public static void main(String args[])throws IOException{

        BufferedReader in;

        in=new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Introduzca un numero entero positivo");
```

```

int a=Integer.parseInt(in.readLine());

if(a%2==0)

System.out.println("El numero es par");

else

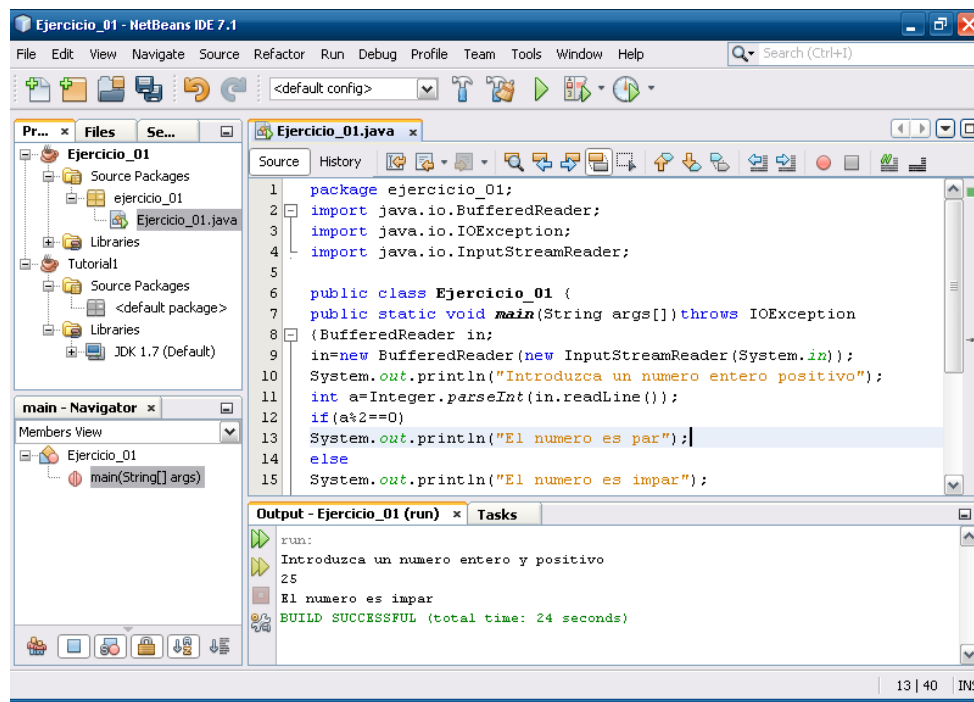
System.out.println("El numero es impar");

}

}

```

Se verá el código de la siguiente manera:



Resultado: Como se observa en el apartado de salida de Netbeans, el programa muestra la salida de la ejecución.

2. Dado un número entero que se introduce por teclado, determinar si es positivo, negativo o nulo.

EXPLICACION DEL PROGRAMA: El usuario ingresa un numero entero y el programa determina si es positivo, negativo o cero(nulo). Utiliza simples comparaciones y sale el mensaje correspondiente.

```
package ejercicio_02;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Ejercicio_02 {

    public static void main (String args[])throws IOException{

        BufferedReader in;

        in = new BufferedReader (new InputStreamReader (System.in));

        System.out.println("Ingrese un numero entero: ");

        int d = Integer.parseInt(in.readLine());

        if (d==0)

            System.out.print(" el numero es nulo ");
```

```
else
```

```
{if (d<0)
```

```
System.out.print("El numero es negativo");
```

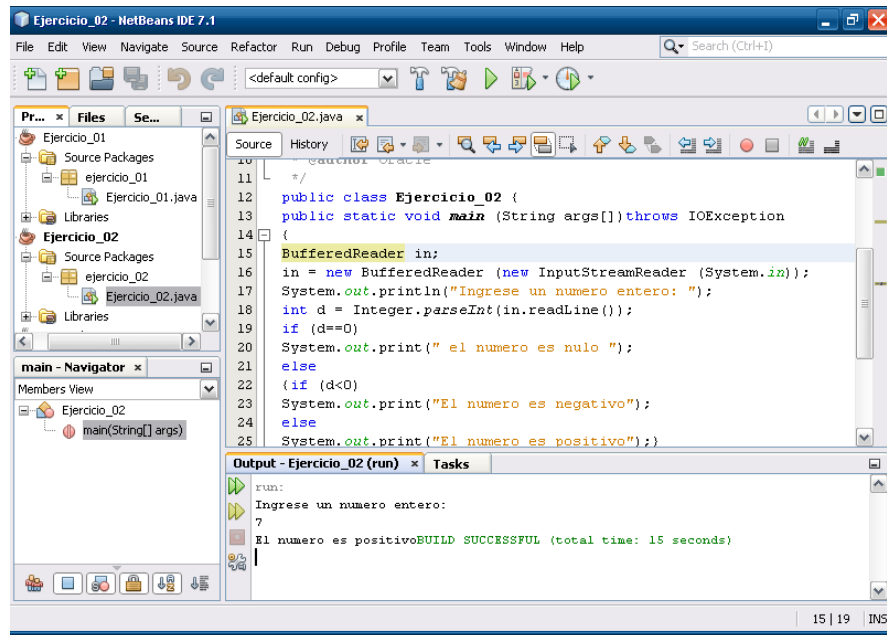
```
else
```

```
System.out.print("El numero es positivo");}
```

```
}
```

```
}
```

Se verá el código de la siguiente manera:



Resultado: Como se observa en el apartado de salida de Netbeans, el programa muestra la salida de la ejecución.

3. Dado un número entero que se introduce por teclado, determinar si se encuentra en el intervalo cerrado 51 - 100.

EXPLICACIÓN DEL PROGRAMA: Este programa nos permite Verificar que un número ingresado por teclado se encuentre en el intervalo cerrado [51,100]

```
package ejercicio_03;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Ejercicio_03 {

    public static void main (String args[])throws IOException{

        BufferedReader in;

        in = new BufferedReader (new InputStreamReader (System.in));

        int n;

        System.out.println("Ingrese el numero a verificar: ");

        n = Integer.parseInt(in.readLine());

        if(n>=51&n<=100)
```

```

        System.out.println("El numero se encuentra en el intervalo cerrado
[51-100]");

    else

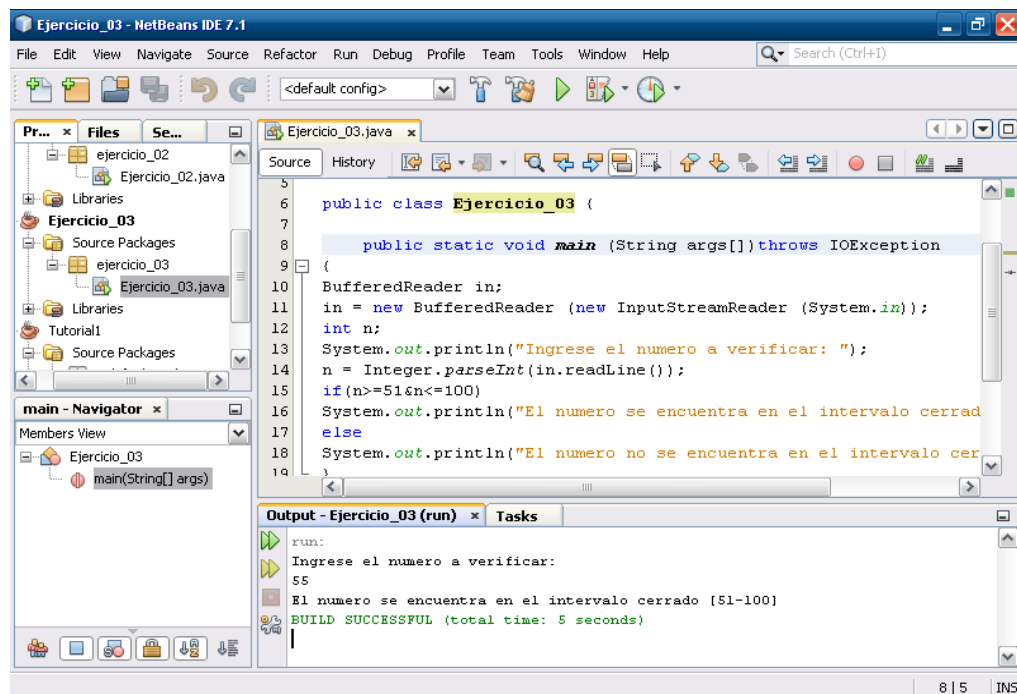
        System.out.println("El numero no se encuentra en el intervalo cerrado
[51-100]");

    }

}

```

Se verá el código de la siguiente manera:



Resultado: Como se observa en el apartado de salida de Netbeans, el programa muestra la salida de la ejecución.

4. Dado un número entero que se introduce por teclado, determinar si es negativo o superior a 100.

EXPLICACION DEL PROGRAMA: el programa en principio define la clase Ejercicio_04 que contiene la función principal main. Primero se lee un numero, que es introducido por teclado y se hacen comparaciones para verificar si es negativo, mayor a 100 o en el intervalo 0-100. Finalmente se despliega por pantalla la condición de dicho número.

```
package ejercicio_04;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Ejercicio_04 {

    public static void main(String args[])throws IOException{

        BufferedReader in;

        in=new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Introduzca un numero");

        int a=Integer.parseInt(in.readLine());

        if(a<0)
```

```

System.out.println("El numero es negativo");

else

if (a>100)

System.out.println("El numero es mayor a 100");

else

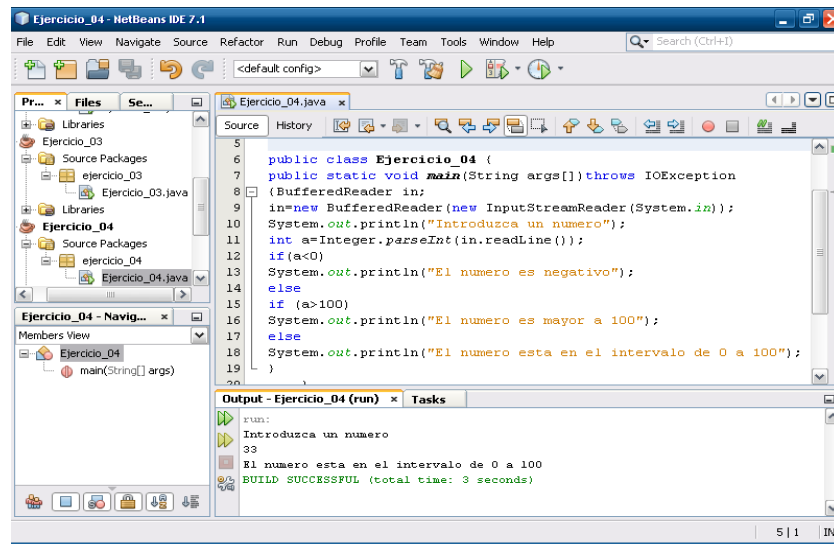
System.out.println("El numero esta en el intervalo de 0 a 100");

}

}

```

Se verá el código de la siguiente manera:



Resultado: Como se observa en el apartado de salida de Netbeans, el programa muestra la salida de la ejecución.

5. Dado un número que se introduce por teclado, si es positivo verificar si se encuentra en el intervalo abierto 60 – 90, de lo contrario emitir un mensaje de error.

EXPLICACION DEL PROGRAMA: Este programa verifica si un número ingresado por el teclado es positivo y si se encuentra en un intervalo predeterminado utilizando sentencias if.

```
package ejercicio_05;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Ejercicio_05 {

    public static void main(String args[])throws IOException{

        BufferedReader in;

        in=new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Introduzca un numero");

        int a=Integer.parseInt(in.readLine());

        if(a<61)
```

```
System.out.println("ERROR");
```

```
else
```

```
if (a>89)
```

```
System.out.println("ERROR");
```

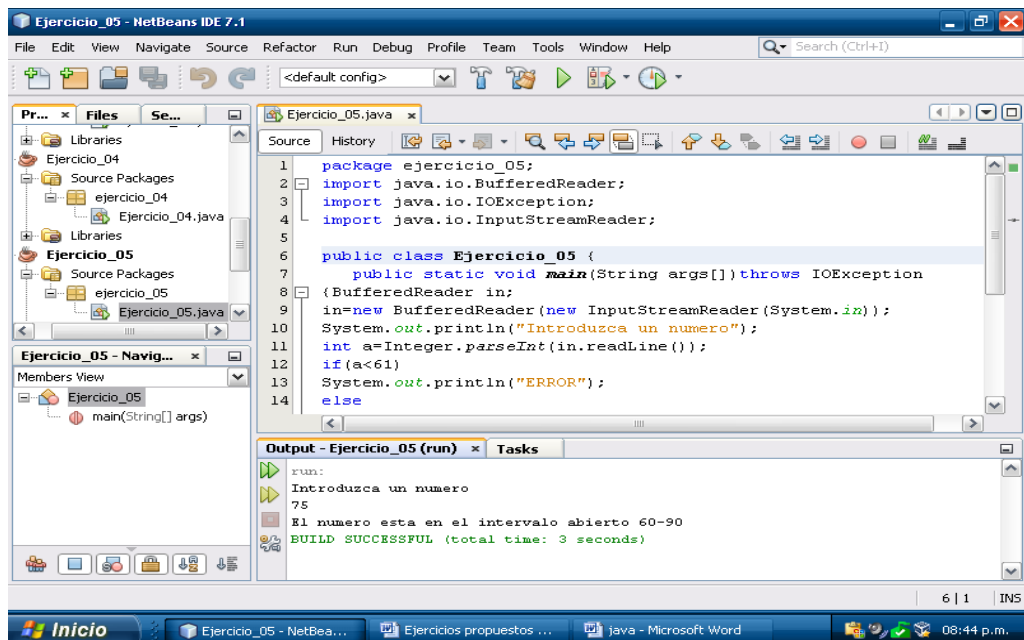
```
else
```

```
System.out.println("El numero esta en el intervalo abierto 60-90");
```

```
}
```

```
}
```

Se verá el código de la siguiente manera:



Resultado: Como se observa en el apartado de salida de Netbeans, el programa muestra la salida de la ejecución.

6. Una fuente de datos registra varias edades, la edad 0 indica el final del ingreso de datos, realice un programa para determinar el promedio de las edades ingresadas y además el porcentaje de personas mayores a los 50 años.

EXPLICACIÓN DEL PROGRAMA: Este programa nos permite registrar varias edades con la información que una vez ingresada la edad 0 esta nos indica el final del ingreso de edades, calculando le promedio de las edades y además el porcentaje de edades mayores a los 50 años.

```
package ejercicio_06;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Ejercicio_06 {

    public static void main (String args []) throws IOException{

        BufferedReader in;

        in = new BufferedReader (new InputStreamReader (System.in));

        double m = 1, aux = 0, n = 0, o = 0;

        System.out.println ("La edad 0 indica el final de ingreso de edades");
```

```

while (m != 0){

    System.out.print ("Ingrese una edad ");

    m = Integer.parseInt (in.readLine ());

    n++;

    aux = aux + m;

    if (m > 50){

        o++;

    } }

    double s = (o * 100) / (n - 1);

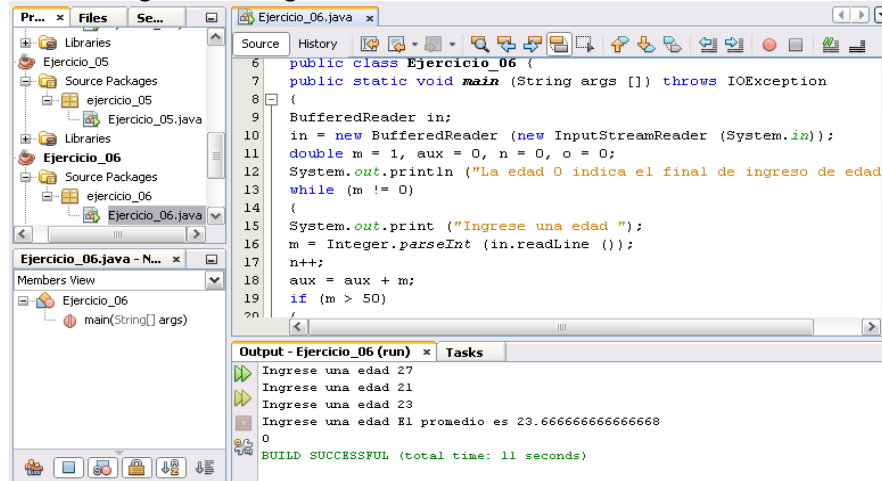
    double aux2 = aux / (n - 1);

    System.out.println ("El promedio es " + aux2 );

} }

```

Se verá el código de la siguiente manera:



Resultado: Como se observa en el apartado de salida de Netbeans, el programa muestra la salida de la ejecución.

7. Leer una cierta cantidad de estaturas (detenga el proceso mediante un centinela) y determine el promedio de estaturas por debajo de 1,60 mts. y el promedio de estaturas en general.

EXPLICACION DEL PROGRAMA: Este programa lee estaturas hasta encontrar un 0 (el sentinel), luego saca un promedio de las estaturas bajo los 1.60m y un promedio de todas las estaturas.

```
package ejercicio_07;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Ejercicio_07 {

    public static void main(String arg[])throws IOException

    {BufferedReader    in    =    new    BufferedReader(new

    InputStreamReader(System.in));

    double []x=new double[100];

    double a=1,pb=0,pg=0,nb=0;

    int i =0;
```

```
System.out.println("Introduzca las estaturas (0 para terminar):");
```

```
a=(Float.valueOf (in.readLine ())).floatValue ();
```

```
if (a>=0)
```

```
{x[i]=a;
```

```
i++;}
```

```
while (a!=0)
```

```
{a=(Float.valueOf (in.readLine ())).floatValue ();
```

```
x[i] =a;
```

```
pg=pg+x[i];
```

```
if (x[i]<1.6)
```

```
{pb=pb+x[i];
```

```
nb++;}
```

```
i++;}
```

```
i=i-2;
```

```
System.out.println("El promedio de estaturas menores a 1.60m es:
```

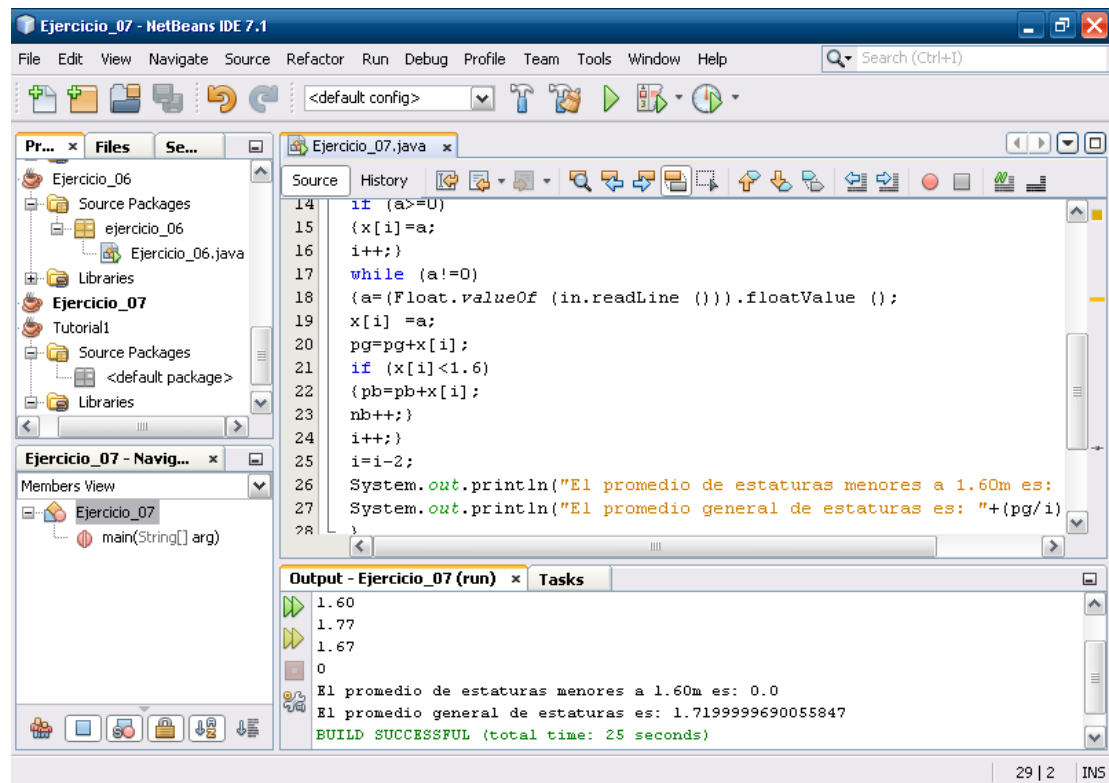
```
"+(pb/nb));
```

```
System.out.println("El promedio general de estaturas es: "+(pg/i));
```

```
}
```

```
}
```

Se verá el código de la siguiente manera:



Resultado: Como se observa en el apartado de salida de Netbeans, el programa muestra la salida de la ejecución.

2011

Universidad Tecnológica
de El Salvador

Gerson Gerardo Argueta

[GUÍA DE PROGRAMACIÓN ORACLE Y JAVA WEB]

A continuación una serie de ejercicios para la programación de un sitio web utilizando Oracle 11G y Java WEB (JSP, Servlet's)

Índice

[Software](#)

[Configuraciones](#)

[Proyecto WEB JSP](#)

[Uso de JNDI para utilizar un pool de conexiones](#)

[Pasos para configurar JNDI](#)

[Utilizando JavaServer Pages Standard Tag Library JSTL](#)

[Pasos para configurar JSTL](#)

[Ejemplo 2 Consultas utilizando JNDI y JSTL](#)

[Ejemplo 3 Consultas utilizando Servlet](#)

[Ejemplo 4 Insert, update utilizando JNDI/JSTL](#)

[Ejemplo 5 Uso de Procedimientos almacenados desde Servlet's](#)

Software

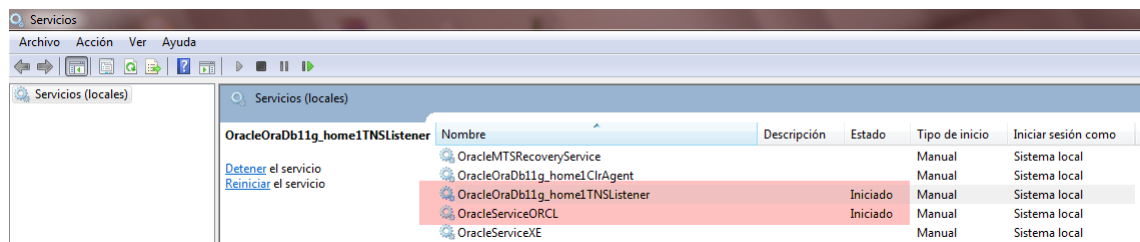
Para el desarrollo de esta guía se debe instalar el software detallado a continuación:

- Netbeans 7.0 (<http://netbeans.org/>)
- Driver JDBC Para Oracle
(<http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html>)
- Servidor Apache Tomcat 7 <http://tomcat.apache.org/>
- Servidor Oracle 11G
<http://www.oracle.com/technetwork/database/enterprise-edition/overview/index.html>

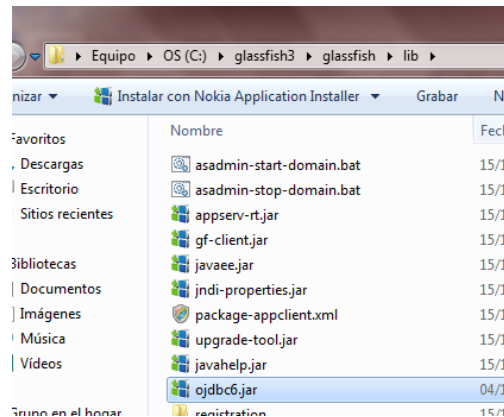
Configuraciones

Configuraciones necesarias para desarrollar páginas web JSP con conexión a BD Oracle

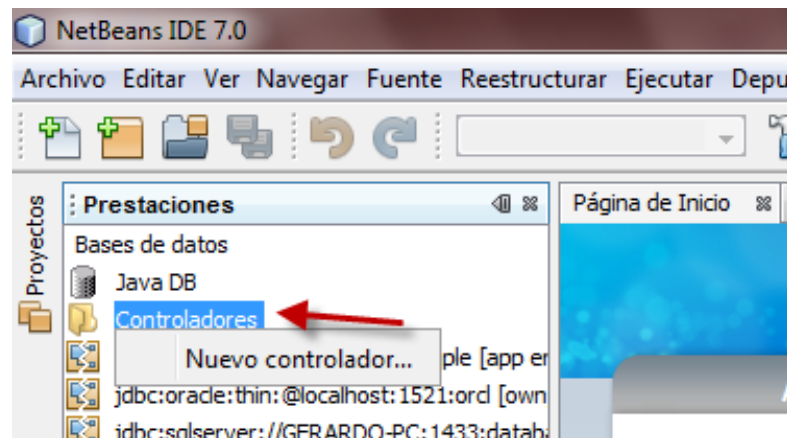
1. Asegurarse que estén habilitados los servicios de Oracle



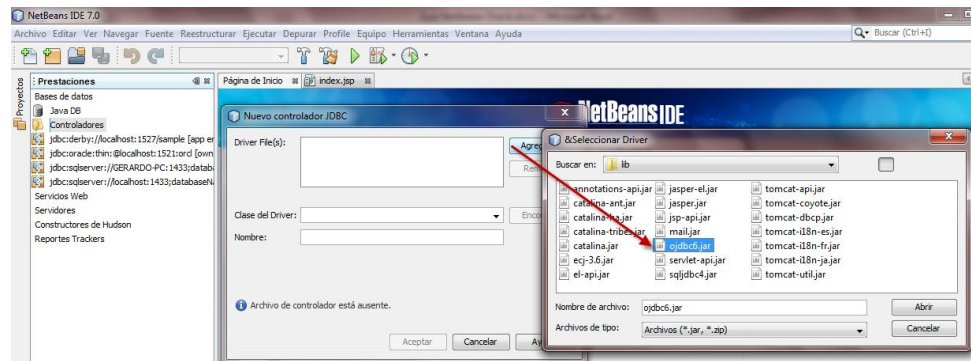
2. Copiar driver jdbc para Oracle en la carpeta de librerías del servidor GlassFish



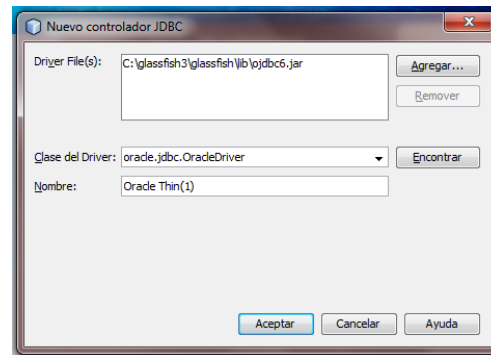
3. Accesar a Netbeans y configurar conexión a Oracle
 - a. En la ventana presentaciones dar clic derecho a la carpeta controladores para agregar uno nuevo



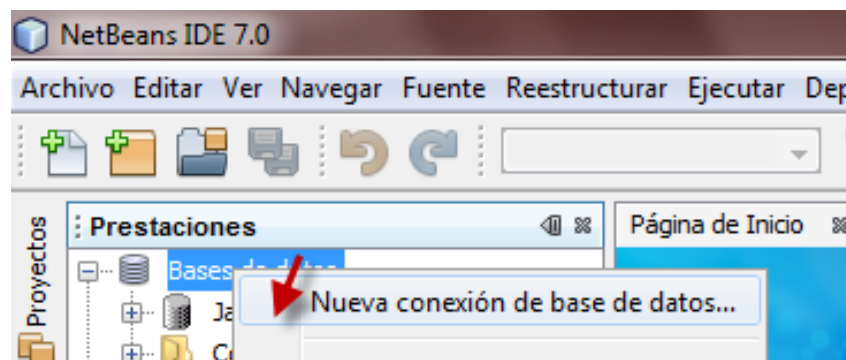
- b. Seleccionar el driver ojdbc6.jar y dar clic al botón aceptar



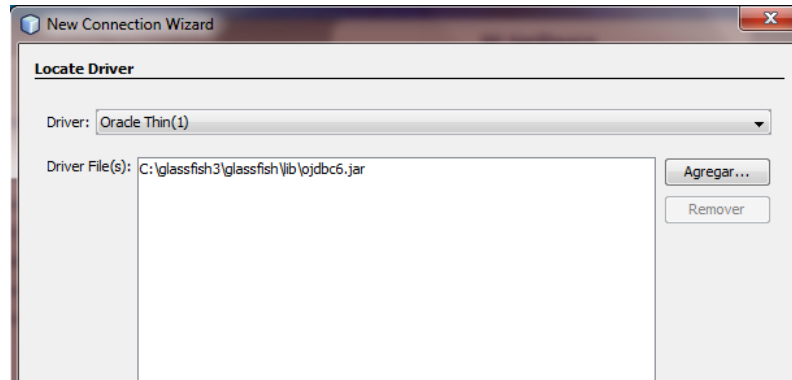
c. Seleccionar el driver ojdbc6.jar y dar clic al botón aceptar



d. Crear nueva conexión. Clic derecho a Bases de Datos



e. Seleccionar driver de Oracle y dar clic al botón siguiente



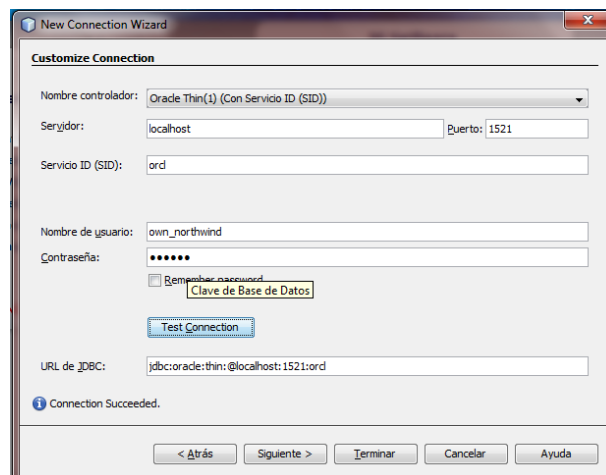
- f. Configurar la conexión, verificar con Test Connection, luego dar clic al botón siguiente

Servidor = localhost

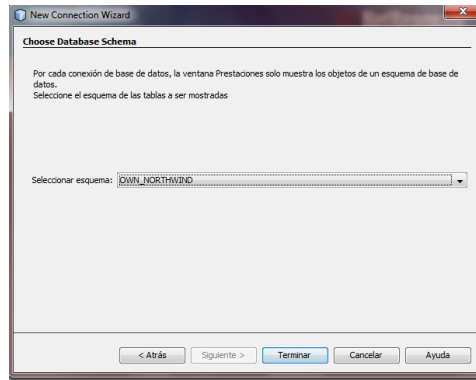
Puerto= 1521

Servicio= orcl

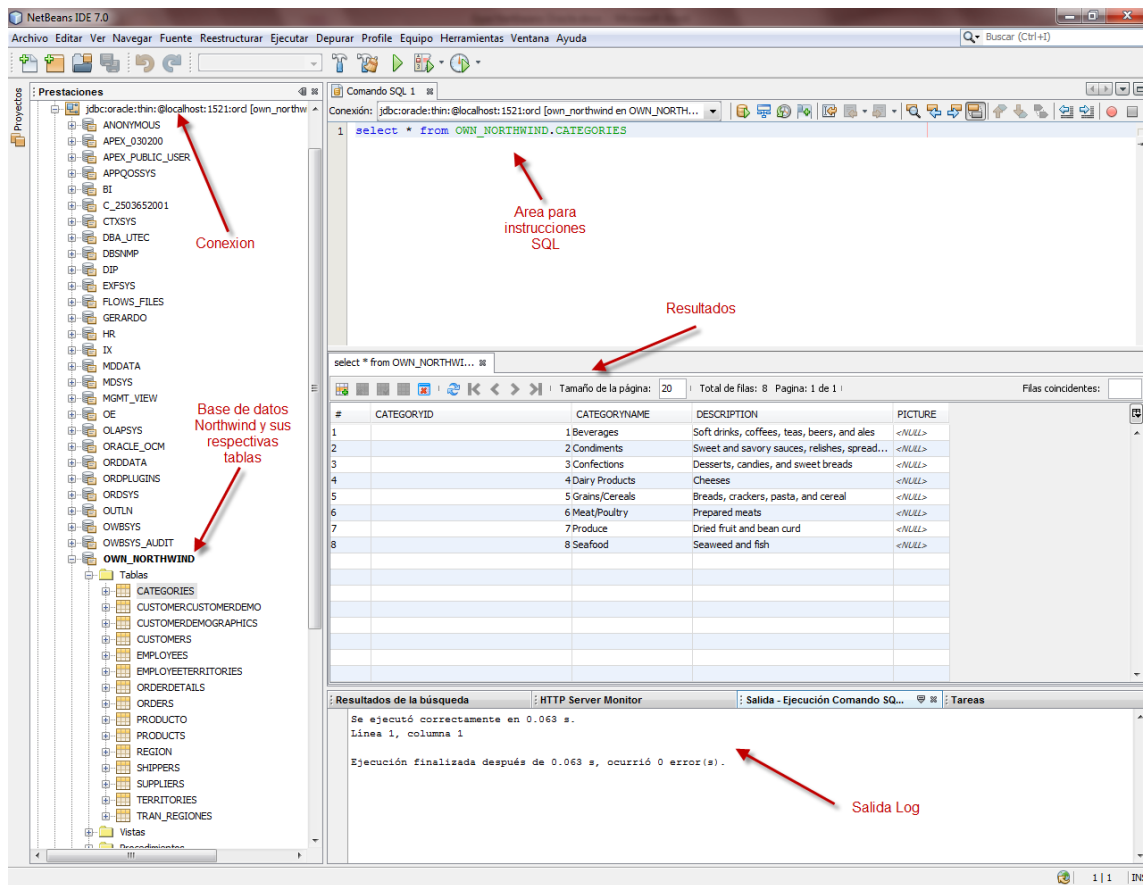
Usuario= own_northwind



- g. Seleccionar el esquema y terminar la configuración



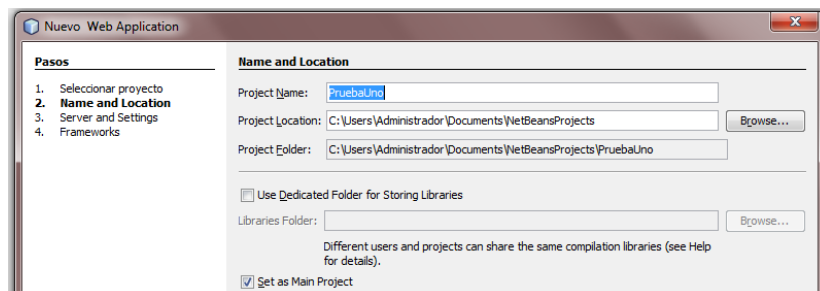
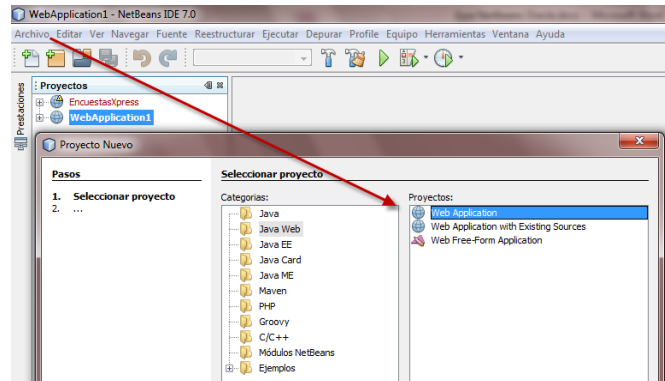
- h. En la ventana de presentaciones seleccionar la conexión creada y verificar las tablas en la base de datos, también se pueden efectuar instrucciones sql desde Netbeans



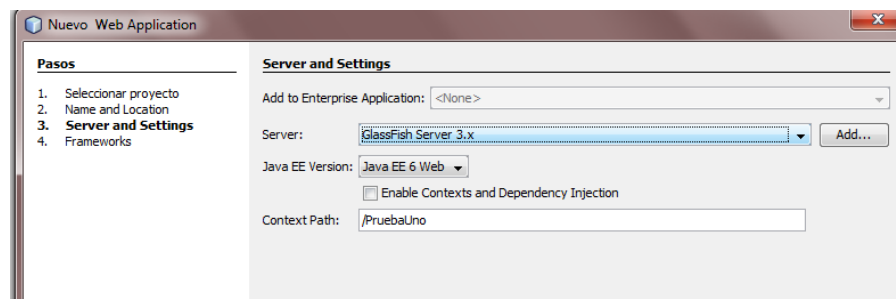
Proyecto WEB JSP

A continuación los pasos para la creación de un proyecto web JSP

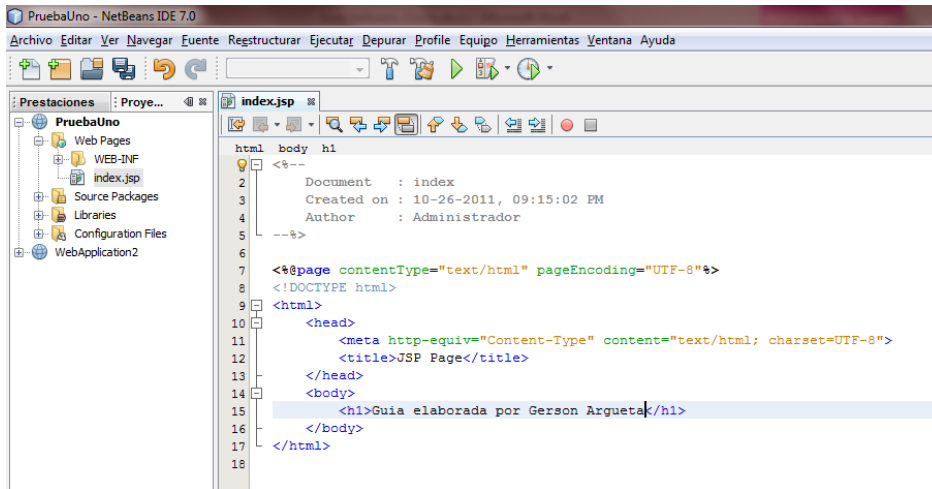
1. Selección nuevo proyecto, Java Web , Web Application



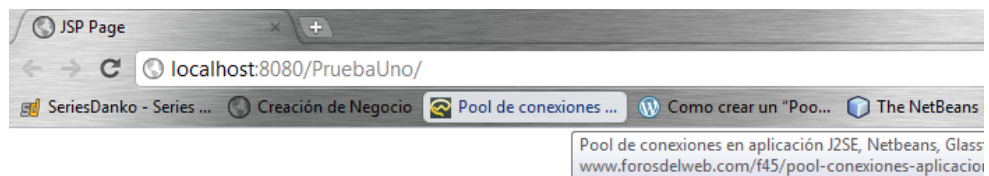
2. Selección Server GlassFish Server 3.x



3. Una vez creado el proyecto , probar su funcionamiento con F6

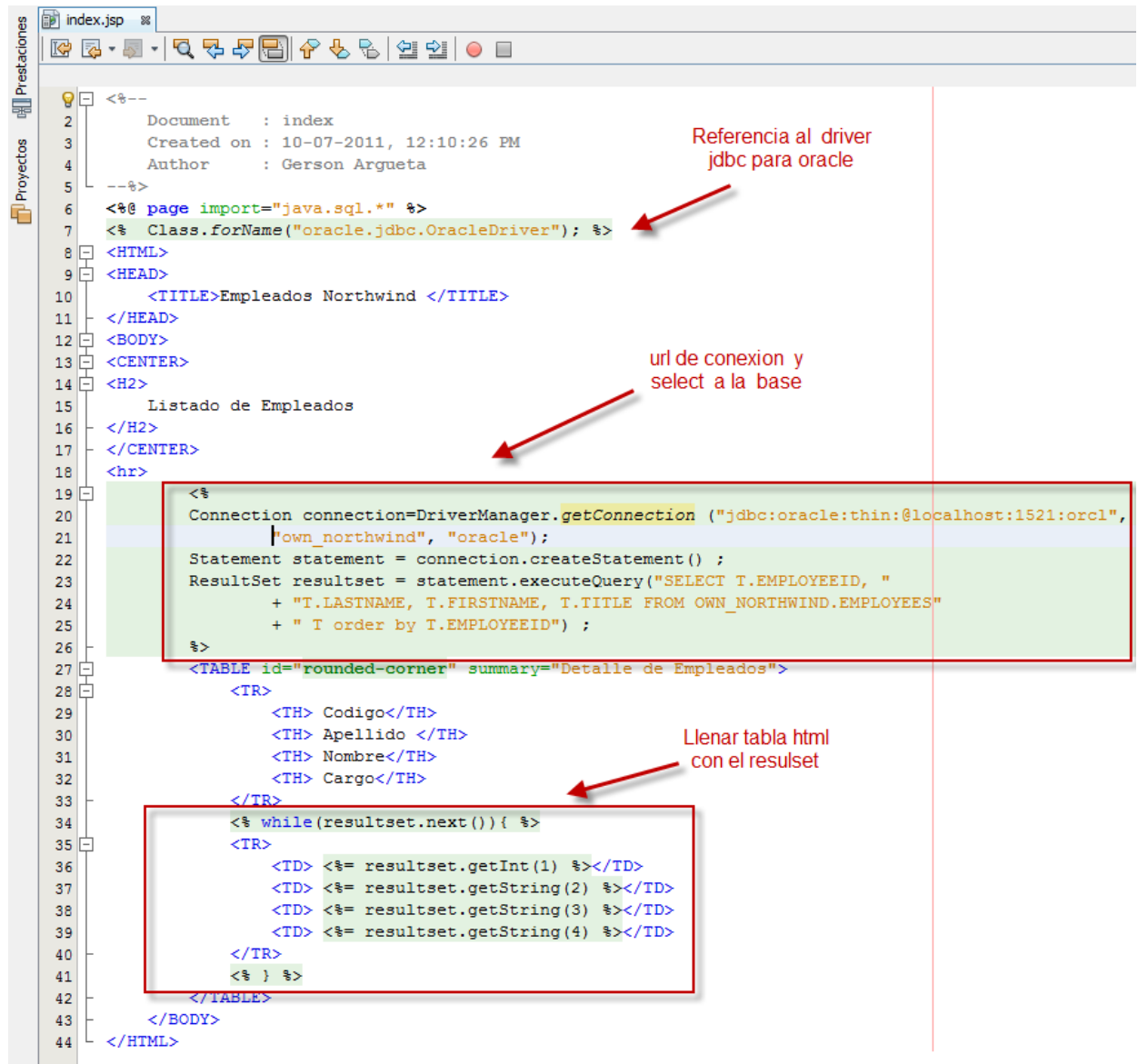


4. Se debe de cargar la página en el explorador de internet



Guia elaborada por Gerson Argueta

Ejemplo 1 Consulta efectuada desde la página JSP. Digitar el siguiente código en la página index.jsp



The screenshot shows an IDE window titled 'index.jsp'. The code is as follows:

```
1 <!--
2 Document : index
3 Created on : 10-07-2011, 12:10:26 PM
4 Author : Gerson Argueta
5 -->
6 <%@ page import="java.sql.*" %>
7 <% Class.forName("oracle.jdbc.OracleDriver"); %>
8
9 <HTML>
10 <HEAD>
11 <TITLE>Empleados Northwind </TITLE>
12 </HEAD>
13 <BODY>
14 <CENTER>
15 <H2>
16 Listado de Empleados
17 </H2>
18 </CENTER>
19 <hr>
20
21 <%
22 Connection connection=DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:orcl",
23 |'own_northwind", "oracle");
24 Statement statement = connection.createStatement() ;
25 ResultSet resultSet = statement.executeQuery("SELECT T.EMPLOYEEID, "
26 + "T.LASTNAME, T.FIRSTNAME, T.TITLE FROM OWN_NORTHWIND.EMPLOYEES"
27 + " T order by T.EMPLOYEEID") ;
28 %>
29
30 <TABLE id="rounded-corner" summary="Detalle de Empleados">
31 <TR>
32 <TH>Codigo</TH>
33 <TH>Apellido </TH>
34 <TH>Nombre</TH>
35 <TH>Cargo</TH>
36 </TR>
37
38 <% while(resultSet.next()) { %>
39 <TR>
40 <TD> <%= resultSet.getInt(1) %></TD>
41 <TD> <%= resultSet.getString(2) %></TD>
42 <TD> <%= resultSet.getString(3) %></TD>
43 <TD> <%= resultSet.getString(4) %></TD>
44 </TR>
45 <% } %>
46
47 </TABLE>
48 </BODY>
49 </HTML>
```

Annotations with red arrows:

- Line 7: **Referencia al driver jdbc para oracle**
- Line 22: **url de conexion y select a la base**
- Line 38: **Llenar tabla html con el resulset**

Resultado



Codigo	Apellido	Nombre	Cargo
1	uuu	uuu	uu
2	Fuller	Andrew	Vice President, Sales
3	Leverling	Janet	Sales Representative
4	Peacock	Margaret	Sales Representative
5	Buchanan	Steven	Sales Manager
6	Suyama	Michael	Sales Representative
7	King	Robert	Sales Representative
8	Callahan	Laura	Inside Sales Coordinator
9	Dodsworth	Anne	Sales Representative
51	PEREZ	CARLOS	CONTADOR
52	TURCIOS	JUAN	ASISTENTE
53	PEREZ	CARLOS	CONTADOR
55	HERNANDEZ	GERARDO	SUPERVISOR
56	LEMUS	VIVIANA	OPTOMETRISTA
57	ARGUETA	DANIELA	SECRETARIA
58	CAROLINA	ARIANA	CONTADOR
59	fdfd	ffd	dfdf
79	argueta	gerson	prueba
80	PEREZ	CARLOS	PRUEBA

Uso de JNDI para utilizar un pool de conexiones

(<http://tomcat.apache.org/tomcat-5.5-doc/jndi-datasource-examples-howto.html>)

Concepto de pool de conexiones

Una aproximación elemental a JDBC implica que se realiza una conexión a la base de datos en cada servlet. Se repite el esquema conexión-operación-desconexión. Esta forma de trabajar es perfectamente válida, pero resulta ineficiente, ya que se están desperdiciando ciclos de ejecución en cada conexión y desconexión. Existe otra alternativa: crear un pool de conexiones, es decir, mantener un conjunto de conexiones. Cuando un servlet/JSP requiere una conexión la solicita al conjunto de conexiones disponibles (desocupadas, "idle") y cuando no la va a usar la devuelve al pool. De esta forma nos ahorramos el consumo de tiempo de conexión y desconexión.

Pasos para configurar JNDI

Existen dos formas de crear un pool de conexiones:

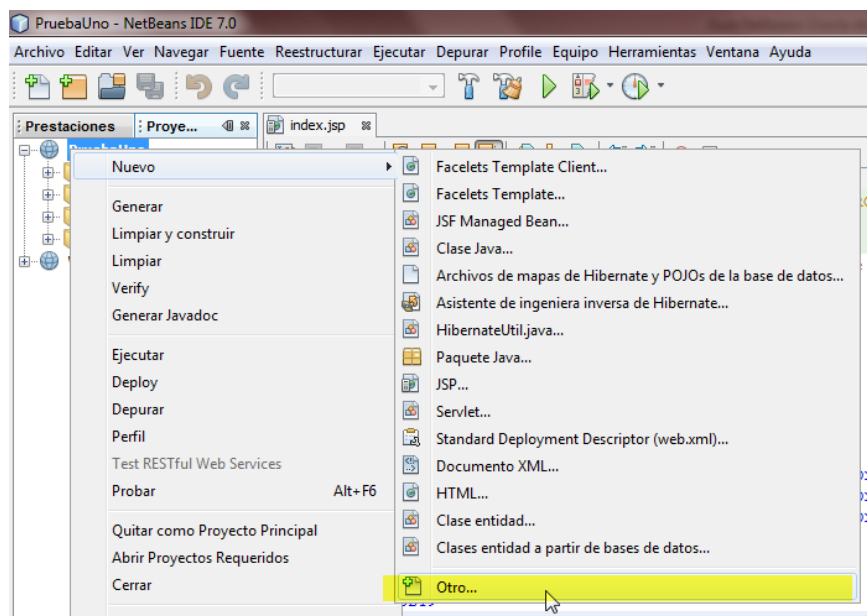
1. Crear la conexión desde un proyecto de Netbeans
2. Crear la conexión en la consola de administración del servidor GlassFish

Pool de conexiones creadas en Netbeans

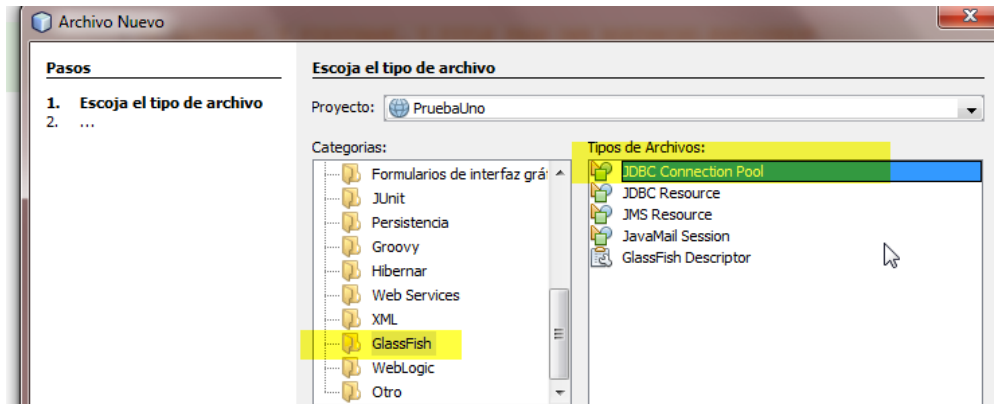
Esta forma es recomendable para entornos de desarrollo ya que se crean de forma fácil.

Al proyecto recién creado, agregar un JDBC Connection Pool

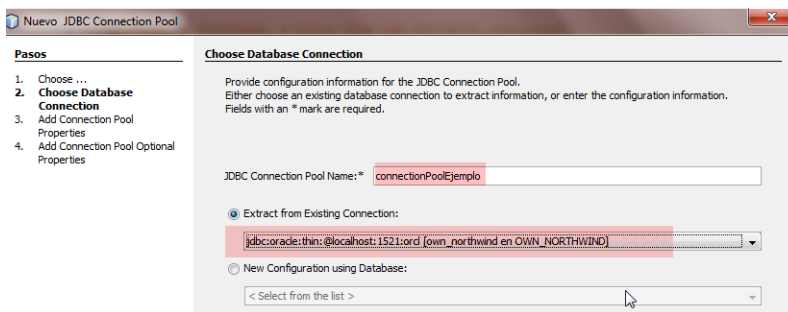
Seleccionar Nuevo/Otro...



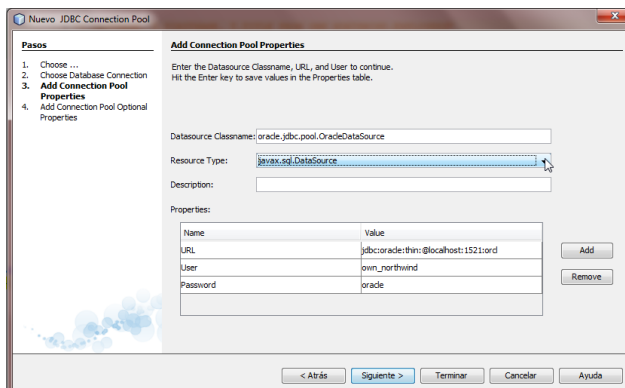
Seleccionar la opción JDBC Connection Pool de la carpeta GlassFish y hacer clic en Siguiente



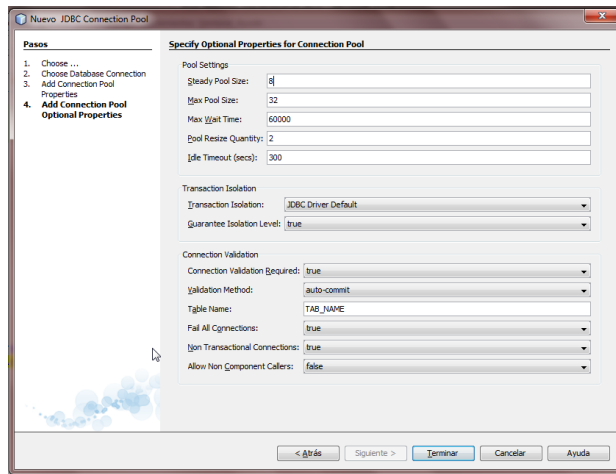
Digitar un nombre en JDBC Connection Pool Name y seleccionar la conexión con Oracle creada en la sección 1



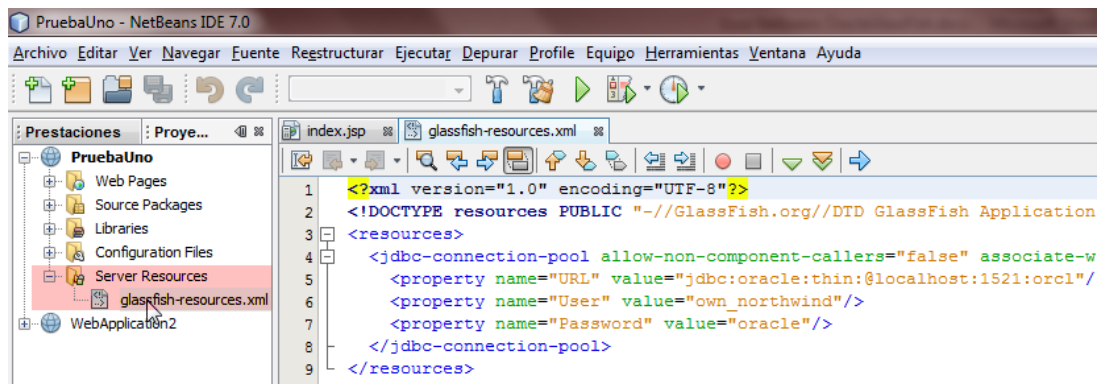
Verificar los datos de la conexión



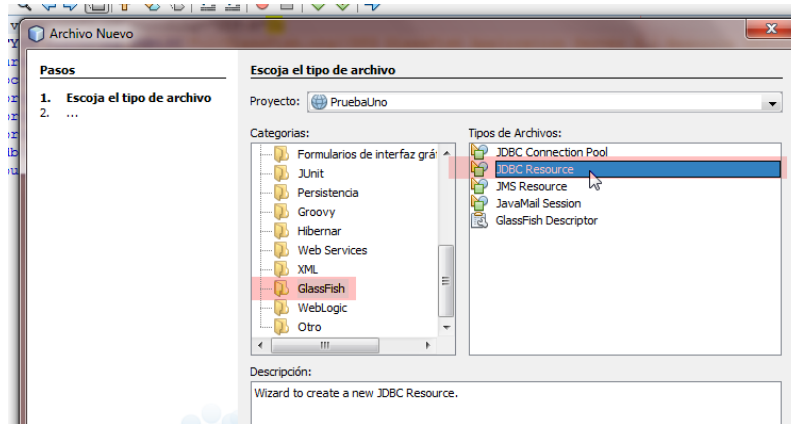
La pantalla siguiente efectuar configuraciones opcionales a la conexión creada, no cambiar nada y dar clic al botón terminar.



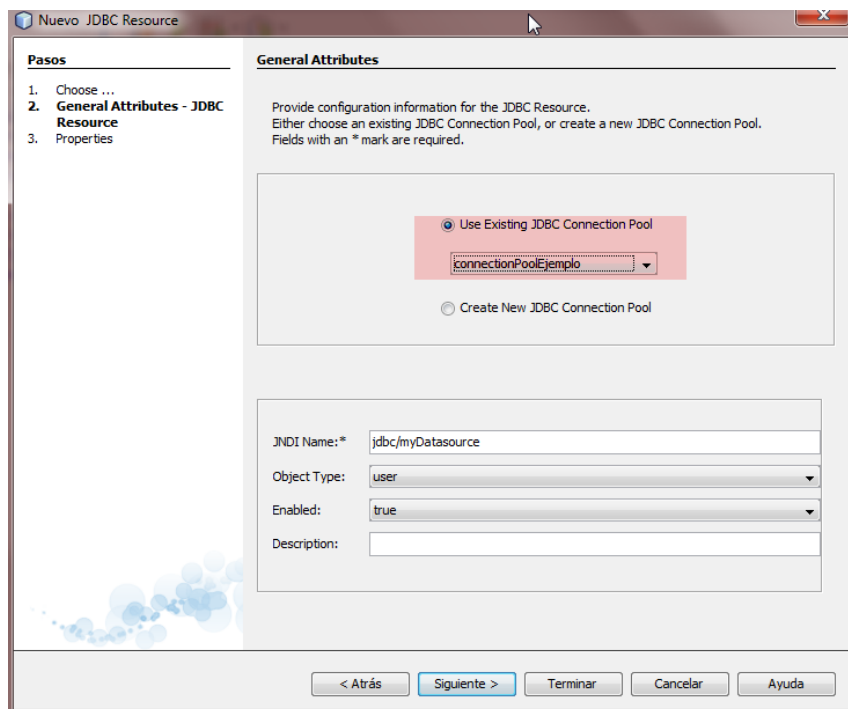
En el proyecto se ha creado una carpeta llamada Server Resources con un archivo de configuración xml.



Adicionar al proyecto un JDBCResource el cual se encuentra en la carpeta GlassFish

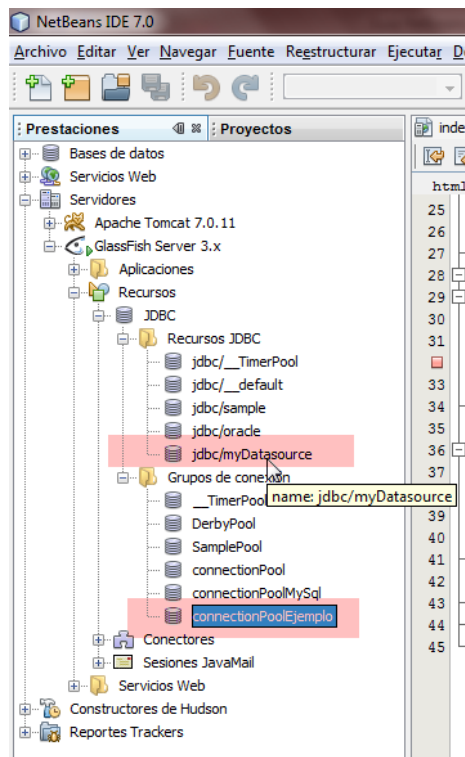


Seleccionar de la lista el pool de conexiones creado anteriormente



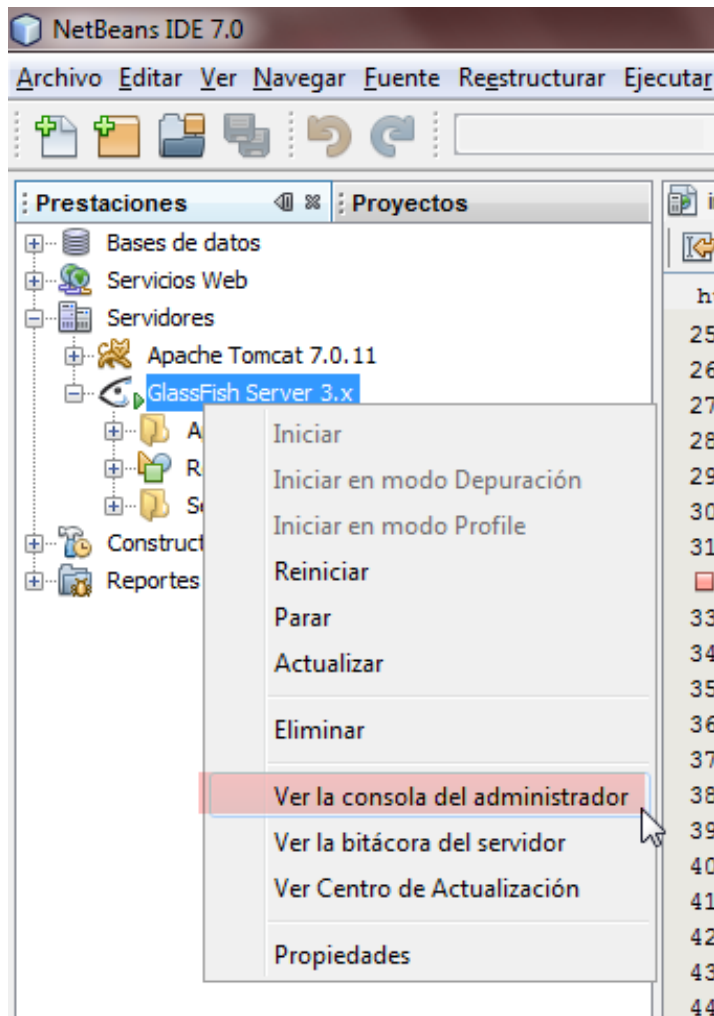
Dar click al botón siguiente y terminar.

Puede verificar la conexión recién creada en la ventana de presentaciones como se muestra en la imagen

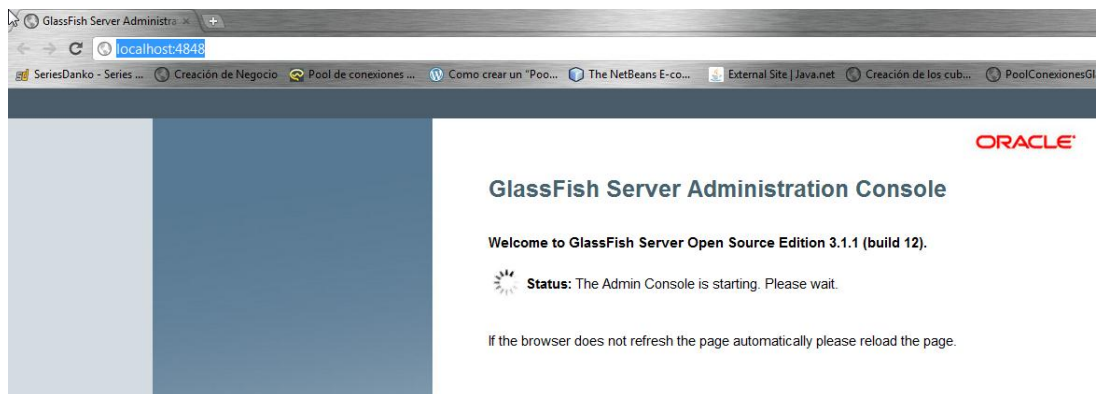


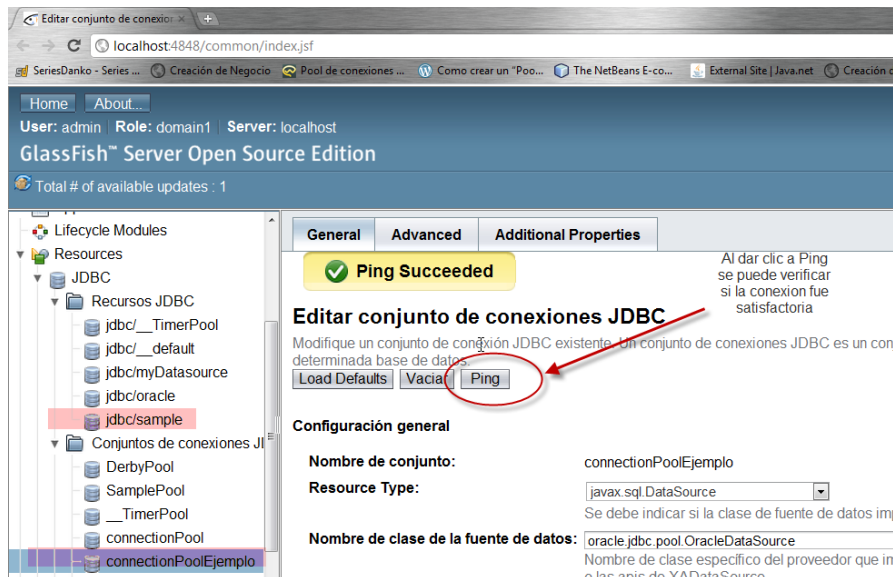
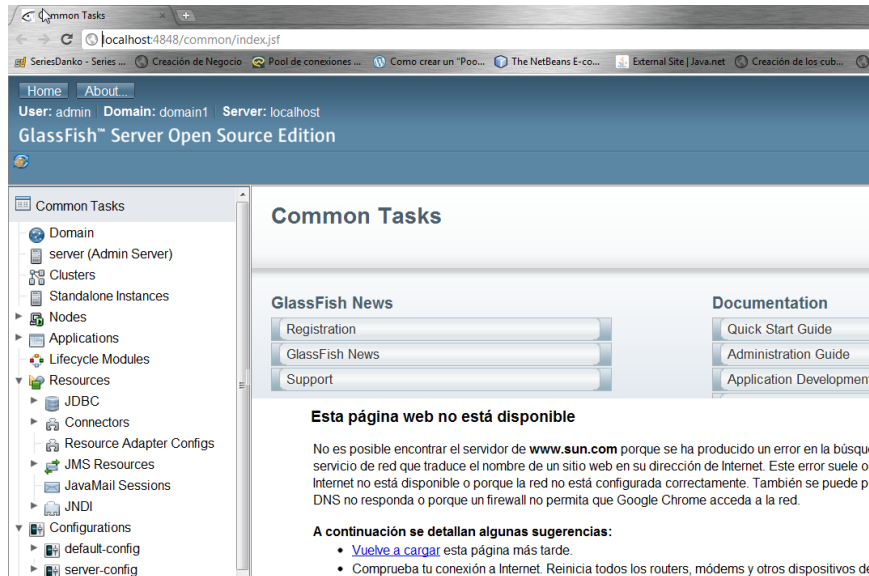
O también accedendo a la consola de administración del servidor. Este paso se puede realizar de dos maneras.

- Clic derecho al servidor en la ventana de presentaciones y seleccionar la opción **Ver la consola del administrador**
- O digitar en el explorador la siguiente dirección <http://localhost:4848>



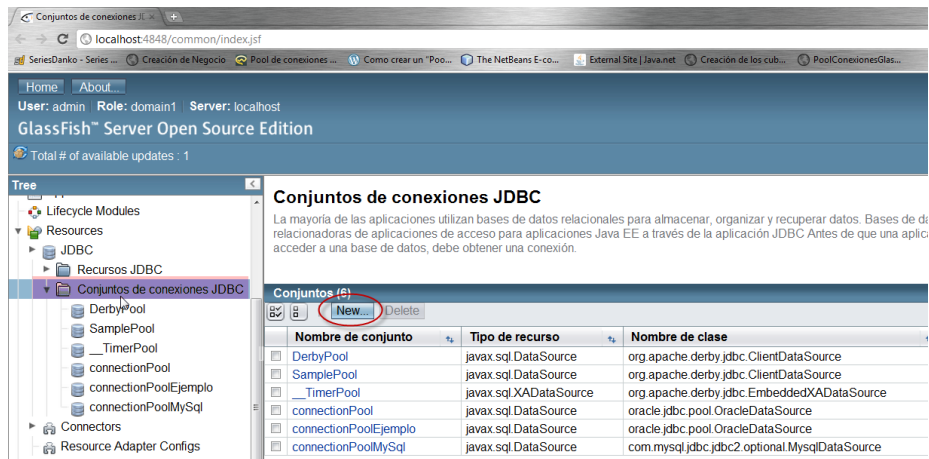
Se iniciara la pantalla de administración





Crear la conexión desde la consola de administración del servidor GlassFish

En la opción Conjunto de Conexiones JDBC dar clic al botón **New**



Configurar según pantalla y dar clic a **next**

Nuevo conjunto de conexiones de JDBC (paso 1 de 2)

Identifique las preferencias generales del conjunto de conexión.

Configuración general

Nombre de conjunto: * ConnPoolDos

Resource Type: javax.sql.DataSource

Se debe indicar si la clase de fuente de datos implementa más de 1 de la interfaz.

Proveedor de los controladores de la base de datos: Oracle

Seleccionar o introducir un proveedor de controladores de la base de datos

Next Cancel

* Indicates required field

En la siguiente pantalla configurar los siguientes parámetros y dar clic a **Finish**

Additional Properties (18)

☒ ☐ | **Add Property** **Delete Properties**

Name	Value	Description:
☐ User	own_northwind	
☐ DatabaseName		
☐ Description		
☐ Password	oracle	
☐ ConnectionCacheName		
☐ ONSConfiguration		
☐ DataSourceName	OracleDataSource	
☐ ServerName		
☐ DriverType		
☐ ExplicitCachingEnabled	false	
☐ MaxStatements	0	
☐ ServiceName		
☐ TNSEntryName		
☐ NetworkProtocol	tcp	
☐ ImplicitCachingEnabled	false	
☐ URL	jdbc:oracle:thin:@localhost:1521:orcl	

Una vez creada seleccionarla del listado y hacer ping para verificarla

Conjuntos de conexiones JDBC

La mayoría de las aplicaciones utilizan bases de datos relacionales para almacenar, organizar y recuperar datos. Las aplicaciones de acceso a bases de datos relacionales utilizan la aplicación JDBC para acceder a una base de datos, debe obtener una conexión.

Conjuntos (7)

☒ ☐ | **New...** **Delete**

Nombre de conjunto	Tipo de recurso	Nombre de clase
ConnPoolDos	javax.sql.DataSource	oracle.jdbc.pool.OracleDataSource

Click to Make This the Primary Sort Column - Ascending

General **Advanced** **Additional Properties**

☒ **Ping Succeeded**

Editar conjunto de conexiones JDBC

Modifique un conjunto de conexión JDBC existente. Un conjunto de conexión determina una base de datos.

Load Defaults **Vaciar** **Ping**

Configuración general

En la opción Recursos JDBC crear un nuevo Recurso y configurar el pool de conexiones creada en el paso anterior



Nuevo recurso JDBC

Especifique un nombre JNDI exclusivo que identifique el recurso JDBC que desea crear. El nombre debe contener alfanuméricos, de subrayado, guiones y puntos.

JNDI Name: *

Pool Name:

Use la página [Conjunto de conexiones de JDBC](#) para crear conjuntos nuevos

Description:

Status: ☒ Enabled

Additional Properties (0)

[Add Property](#) [Delete Properties](#)

Name	Value	Description:
No items found.		

Abrir el archivo glassfish-web.xml y sobrescribir el siguiente código

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE glassfish-web-app PUBLIC "-//GlassFish.org//DTD GlassFish
Application Server 3.1 Servlet 3.0//EN" "http://glassfish.org/dtds/glassfish-web-
app_3_0-1.dtd">
```

<glassfish-web-app error-url="">

<context-root>/PruebaUno</context-root>

<resource-ref>

<res-ref-name>jdbc/oracle</res-ref-name>

<jndi-name>jdbc/oracle</jndi-name>

</resource-ref>

<class-loader delegate="true"/>

<jsp-config>

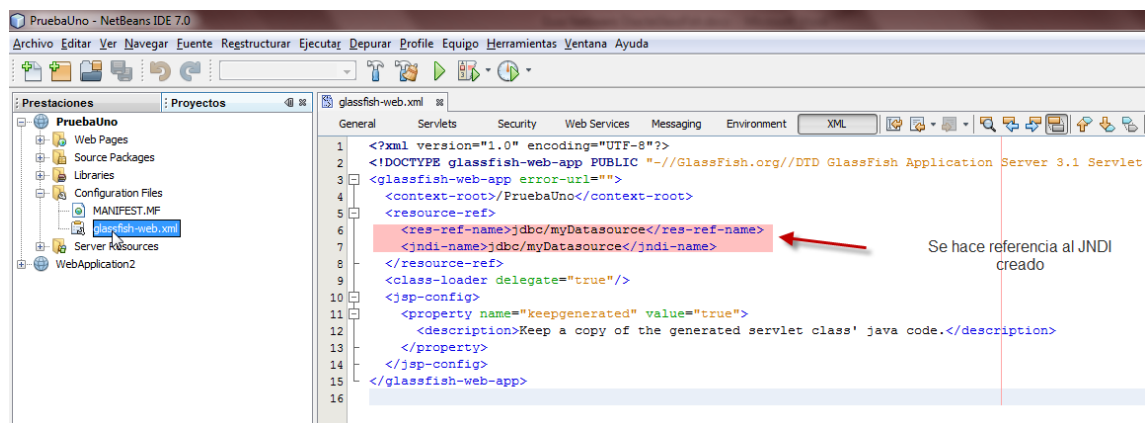
<property name="keepgenerated" value="true">

<description>Keep a copy of the generated servlet class' java code.</description>

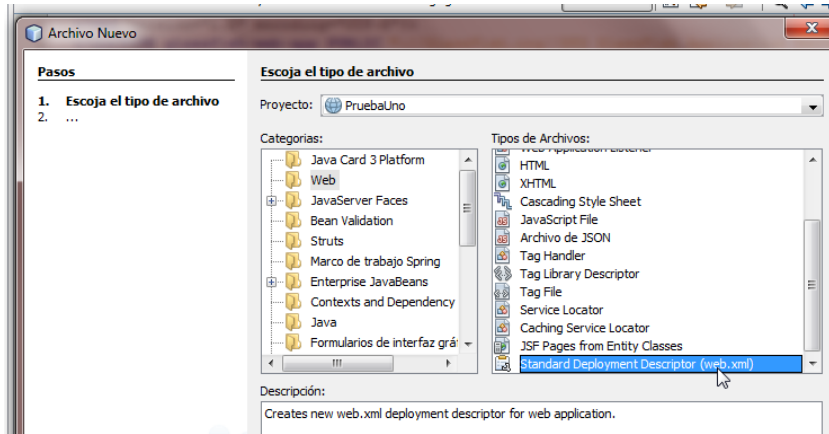
</property>

</jsp-config>

</glassfish-web-app>



Crear y configurar un archivo web.xml



Agregar el siguiente código

```
<description>Es la conexion a sql server</description>
```

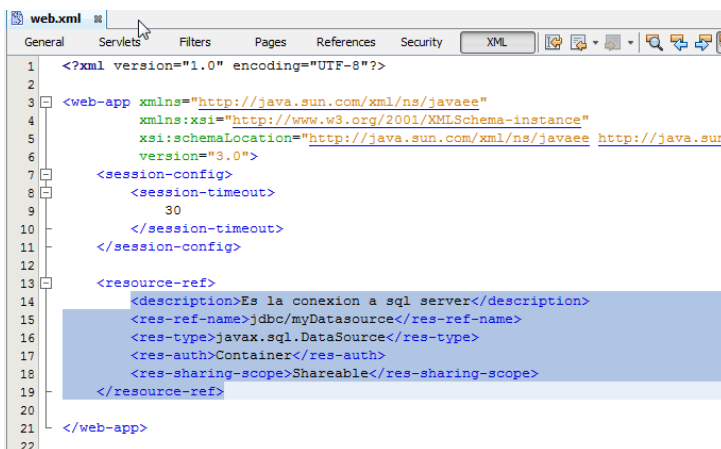
```
<res-ref-name>jdbc/myDatasource</res-ref-name>
```

```
<res-type>javax.sql.DataSource</res-type>
```

```
<res-auth>Container</res-auth>
```

```
<res-sharing-scope>Shareable</res-sharing-scope>
```

```
</resource-ref>
```



Utilizando JavaServer Pages Standard Tag Library JSTL

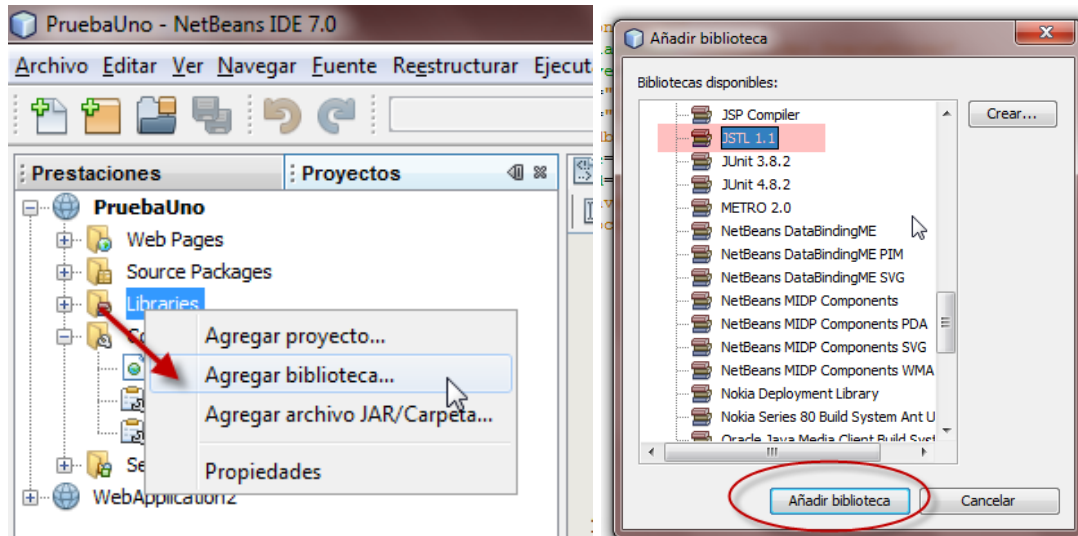
La tecnología JSTL es un componente de Java EE. Extiende las ya conocidas JavaServer Pages (JSP) proporcionando cuatro bibliotecas de etiquetas (Tag Libraries) con utilidades ampliamente utilizadas en el desarrollo de páginas web dinámicas. Estas bibliotecas de etiquetas extienden de la especificación de JSP (la cual a su vez extiende de la especificación de Servlet).

Librería	URI	Prefijo Librería
Core	http://java.sun.com/jsp/jstl/core	c
Internationalization I18N formateo	http://java.sun.com/jsp/jstl/fmt	fmt
SQL/DB support	http://java.sun.com/jsp/jstl/sql	sql
Procesamiento XML	http://java.sun.com/jsp/jstl/xml	x
Functions	http://java.sun.com/jsp/jstl/functions	fn

Pasos para configurar JSTL

Agregar al proyecto la biblioteca JSTL 1.1

Agregar al proyecto las librerías detalladas



Ejemplo 2 Consultas utilizando JNDI y JSTL

Adicionar al proyecto nueva página JSP (SegundoEjemplo.jsp) y digitar el siguiente código

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

    "http://www.w3.org/TR/html4/loose.dtd">

<HTML>

<HEAD>

    <TITLE>Empleados Northwind </TITLE>

</HEAD>

<BODY>
```

<CENTER>

<H2>

Listado de Empleados

</H2>

<H4>

Ejemplo 2 Consultas utilizando JNDI y JSTL

</H4>

</CENTER>

<hr>

<sql:setDataSource var="conexion" dataSource="jdbc/myDatasource" />

<sql:query var="encabezado" dataSource="\${conexion}">

```
SELECT T.EMPLOYEEID, T.LASTNAME, T.FIRSTNAME, T.TITLE
FROM OWN_NORTHWIND.EMPLOYEES T order by T.EMPLOYEEID
```

</sql:query>

<TABLE id="rounded-corner" summary="Detalle de Empleados">

<TR>

<TH>Codigo</TH>

<TH>Apellido </TH>

<TH>Nombre</TH>

<TH>Cargo</TH>

</TR>

<c:forEach var="fila1" items="\${encabezado.rows}">

<TR>

<TD> <c:out value="{fila1.EMPLOYEEID}" /></TD>

<TD> <c:out value="{fila1.LASTNAME}" /></TD>

<TD> <c:out value="{fila1.FIRSTNAME}" /></TD>

<TD> <c:out value="{fila1.TITLE}" /></TD>

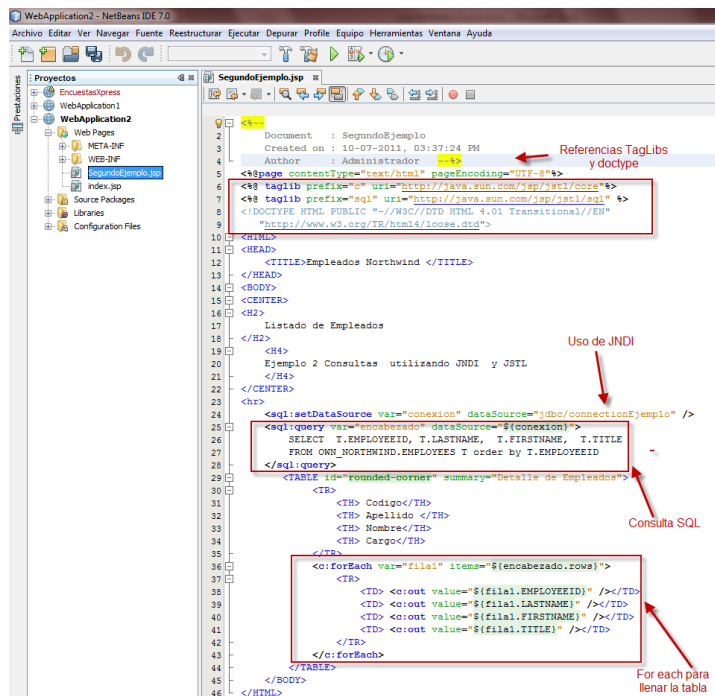
</TR>

</c:forEach>

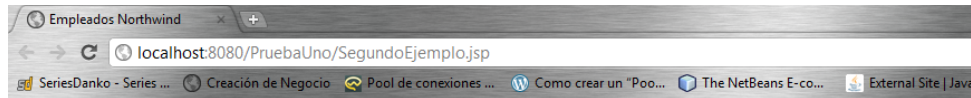
</TABLE>

</BODY>

</HTML>



Ejecutar JSP y verificar resultado



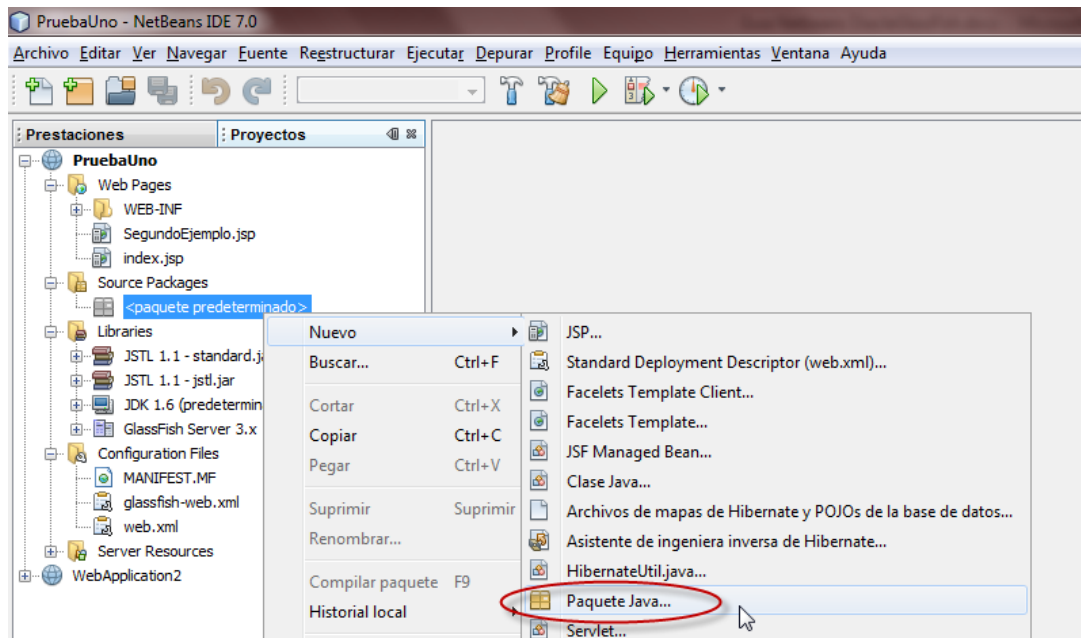
Listado de Empleados

Ejemplo 2 Consultas utilizando JNDI y JSTL

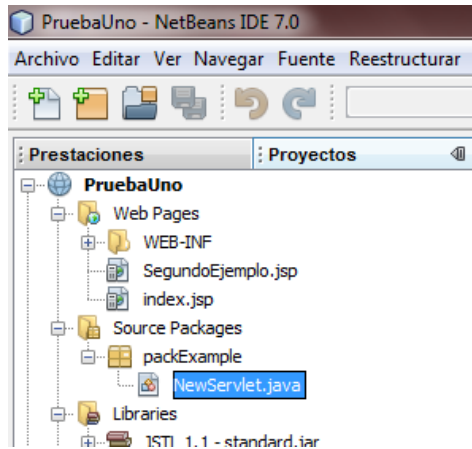
Codigo	Apellido	Nombre	Cargo
1	uuu	uuu	uu
2	Fuller	Andrew	Vice President, Sales
3	Leverling	Janet	Sales Representative
4	Peacock	Margaret	Sales Representative
5	Buchanan	Steven	Sales Manager
6	Suyama	Michael	Sales Representative
7	King	Robert	Sales Representative
8	Callahan	Laura	Inside Sales Coordinator
9	Dodsworth	Anne	Sales Representative
51	PEREZ	CARLOS	CONTADOR
52	TURCIOS	JUAN	ASISTENTE

Ejemplo 3 Consultas utilizando Servlet

En la carpeta Source Packages, agregar nuevo paquete



Agregar nuevo Servlet en el paquete creado



Digitar el siguiente código

```
package packExample;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.naming.NamingException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;
```

```

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.sql.DataSource;

/** * @author Gerson Argueta */

@WebServlet(name = "NewServlet", urlPatterns = {"/NewServlet"})

public class NewServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)

        throws ServletException, IOException, NamingException, SQLException
    {

        response.setContentType("text/html;charset=UTF-8");

        String sql = "SELECT T.EMPLOYEEID, T.LASTNAME, T.FIRSTNAME, "+

            "T.TITLE FROM OWN_NORTHWIND.EMPLOYEES T order by

T.EMPLOYEEID";

        Context ic = new InitialContext();

        DataSource dataSource = (DataSource)

ic.lookup("java:comp/env/jdbc/myDatasource");

        Connection connection = dataSource.getConnection();

        java.sql.Statement stmt = connection.createStatement();

        ResultSet rs = stmt.executeQuery(sql);

        PrintWriter out = response.getWriter();

        try {

```

```

        out.println("<html>");

        out.println("<head>");

        out.println("<title>Ejemplo 3</title>");

        out.println("</head>");

        out.println("<body>");

        out.println("<TABLE><TR> <TH> Codigo</TH> <TH> Apellido </TH>"+"
                + "<TH> Nombre</TH> <TH> Cargo</TH> </TR>");

        while (rs.next()) {

            out.println("<TR><TD>" +rs.getString("EMPLOYEEID")+ "</TD>");

            out.println("<TD>" +rs.getString("LASTNAME")+ "</TD>");

            out.println("<TD>" +rs.getString("FIRSTNAME")+ "</TD>");

            out.println("<TD>" +rs.getString("TITLE")+ "</TD></TR>");

        }

        rs.close();

        out.println("</TABLE> </body>");

        out.println("</html>");

    } finally {

        out.close();

    }

}

```

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

```

/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (NamingException ex) {
        Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (SQLException ex) {
        Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}

```

```

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (NamingException ex) {
        Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (SQLException ex) {
        Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}

```

```

/**
 * Returns a short description of the servlet.
 * @return a String containing servlet description
 */

@Override
public String getServletInfo() {

    return "Short description";

}

```

The screenshot shows a Java code editor with a file named `packExample.java`. The code is a servlet that connects to a database and displays employee information. Red boxes and arrows highlight specific parts of the code:

- Importar librerías:** A red box around the import statements at the top of the file.
- Instanciar objeto de tipo Context, referenciar JNDI, adición de un statement y agregarlo a un resultSet:** A red box around the code that initializes the context, looks up the data source, gets a connection, creates a statement, and executes a query.
- Recorrido del resultSet:** A red box around the `while` loop that iterates through the `ResultSet` and prints the employee data.

```

1 package packExample;
2 import java.io.IOException;
3 import java.io.PrintWriter;
4 import java.sql.Connection;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.util.logging.Level;
8 import java.util.logging.Logger;
9 import javax.naming.Context;
10 import javax.naming.InitialContext;
11 import javax.naming.NamingException;
12 import javax.servlet.ServletException;
13 import javax.servlet.annotation.WebServlet;
14 import javax.servlet.http.HttpServlet;
15 import javax.servlet.http.HttpServletRequest;
16 import javax.servlet.http.HttpServletResponse;
17 import javax.sql.DataSource;
18 /**
19  * @author Gerson Arqueta
20  */
21 @WebServlet(name = "NewServlet", urlPatterns = {"/NewServlet"})
22 public class NewServlet extends HttpServlet {
23     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
24         throws ServletException, IOException, NamingException, SQLException {
25         response.setContentType("text/html; charset=UTF-8");
26         String sql = "SELECT T.EMPLOYEEID, T.LASTNAME, T.FIRSTNAME, "
27             + "T.TITLE FROM OWB.NORTHIND.EMPLOYEES T order by T.EMPLOYEEID";
28         Context ic = new InitialContext();
29         DataSource dataSource = (DataSource) ic.lookup("java:comp/env/jdbc/connectionEjemplo");
30         Connection connection = dataSource.getConnection();
31         java.sql.Statement stmt = connection.createStatement();
32         ResultSet rs = stmt.executeQuery(sql);
33         PrintWriter out = response.getWriter();
34         try {
35             out.println("<html>");
36             out.println("<head>");
37             out.println("<title>Ejemplo 3</title>");
38             out.println("</head>");
39             out.println("<body>");
40             out.println("<table><tr> <th>Codigo</th> <th>Apellido </th>+""
41                 + "<th> Nombre</th> <th> Cargo</th> </tr>");
42             while (rs.next()) {
43                 out.println("<tr><td>" + rs.getString("EMPLOYEEID") + "</td>");
44                 out.println("<td>" + rs.getString("LASTNAME") + "</td>");
45                 out.println("<td>" + rs.getString("FIRSTNAME") + "</td>");
46                 out.println("<td>" + rs.getString("TITLE") + "</td></tr>");
47             }
48             rs.close();
49             out.println("</table> </body>");
50             out.println("</html>");
51         } finally {
52             out.close();
53         }
54     }
55 }

```

Verificar resultado



Codigo	Apellido	Nombre	Cargo
1	uuu	uuu	uu
2	Fuller	Andrew	Vice President, Sales
3	Leverling	Janet	Sales Representative
4	Peacock	Margaret	Sales Representative
5	Buchanan	Steven	Sales Manager
6	Suyama	Michael	Sales Representative

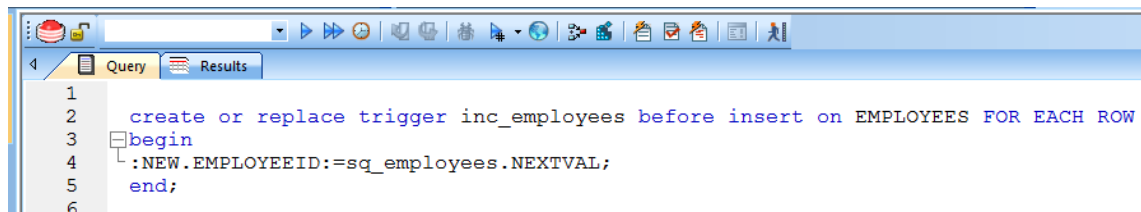
Ejemplo 4 Insert, update utilizando JNDI/JSTL

Adicionar al proyecto nueva página JSP (CuartoEjemplo.jsp). Esta JSP lista detalle de empleados, se adicionan tres link para efectuar las acciones insert, update (se debe de crear otra JSP llamada mantenimientos).

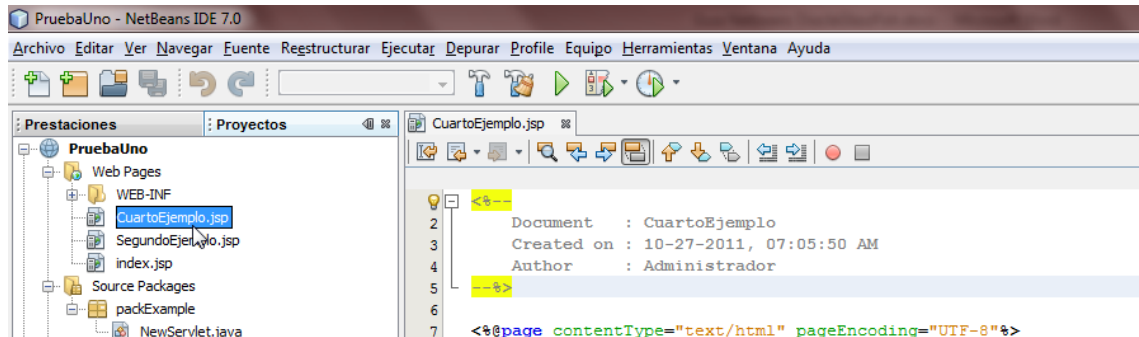
Se debe de crear una secuencia para manejar el código de empleado:

```
create sequence sq_employees start with 1 increment by 1;
```

Crear trigger para que cuando se inserte un valor en la tabla empleados se genere automáticamente el código



Crear CuartoEjemplo.jsp y reemplazar por el siguiente código :



```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
```

```
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Empleados Northwind </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<CENTER>
```

```
<H2>
```

```
Listado de Empleados
```

```
</H2>
```

```
<H4>
```

```
Ejemplo 4 Insert, Update utilizando JNDI y JSTL
```

```

</H4>

</CENTER>

<hr>

<sql:setDataSource var="conexion" dataSource="jdbc/myDatasource" />

<h4 >

    <c:set var="opt" value="I"/>

    <a href="mantenimientos.jsp?opt=${opt}"><strong>Nuevo Empleado ++
</strong></a>

    <c:set var="opt" value="${param.opt}"/>

</h4>

<c:if test="${opt=='I'}">

    <c:set var="VFIRSTNAME" value="${param.FIRSTNAME}"/>

    <c:set var="VLASTNAME" value="${param.LASTNAME}"/>

    <c:set var="VTITLE" value="${param.TITLE}"/>

    <sql:update dataSource="${conexion}" >

        INSERT INTO OWN_NORTHWIND.EMPLOYEES

        (FIRSTNAME, LASTNAME, TITLE)

        VALUES (?, ?, ?)

        <sql:param value="${VFIRSTNAME}"/>

        <sql:param value="${VLASTNAME}"/>

        <sql:param value="${VTITLE}"/>

    </sql:update>

```

```

</c:if>

<c:if test="${opt=='U'}">

    <c:set var="VEMPLOYEEID" value="${param.EMPLOYEEID}"/>

    <c:set var="VFIRSTNAME" value="${param.FIRSTNAME}"/>

    <c:set var="VLASTNAME" value="${param.LASTNAME}"/>

    <c:set var="VTITLE" value="${param.TITLE}"/>

    <sql:update dataSource="${conexion}" >

        UPDATE OWN_NORTHWIND.EMPLOYEES

        SET

        LASTNAME = ?,

        FIRSTNAME=?,

        TITLE=?

        WHERE EMPLOYEEID=?

        <sql:param value="${VLASTNAME}"/>

        <sql:param value="${VFIRSTNAME}"/>

        <sql:param value="${VTITLE}"/>

        <sql:param value="${VEMPLOYEEID}"/>

    </sql:update>

</c:if>

<sql:query var="encabezado" dataSource="${conexion}">

SELECT T.EMPLOYEEID, T.LASTNAME, T.FIRSTNAME, T.TITLE

FROM OWN_NORTHWIND.EMPLOYEES T order by T.EMPLOYEEID

```

```
</sql:query>
```

```
<TABLE id="rounded-corner" summary="Detalle de Empleados">
```

```
<TR>
```

```
<TH>Codigo</TH>
```

```
<TH>Apellido </TH>
```

```
<TH>Nombre</TH>
```

```
<TH>Cargo</TH>
```

```
<TH> </TH>
```

```
</TR>
```

```
<c:forEach var="fila1" items="${encabezado.rows}">
```

```
<TR>
```

```
<TD> <c:out value="${fila1.EMPLOYEEID}" /></TD>
```

```
<TD> <c:out value="${fila1.LASTNAME}" /></TD>
```

```
<TD> <c:out value="${fila1.FIRSTNAME}" /></TD>
```

```
<TD> <c:out value="${fila1.TITLE}" /></TD>
```

```
<td>
```

```
<c:set var="opt" value="U"/>
```

```
<a
```

```
href="mantenimientos.jsp?opt=${opt}&EMPLOYEEID=${fila1.EMPLOYEEID}&LASTNAME=${fila1.LASTNAME}&FIRSTNAME=${fila1.FIRSTNAME}&TITLE=${fila1.TITLE}" class=""> Modificar </a>
```

```
</td>
```

```

</TR>

</c:forEach>

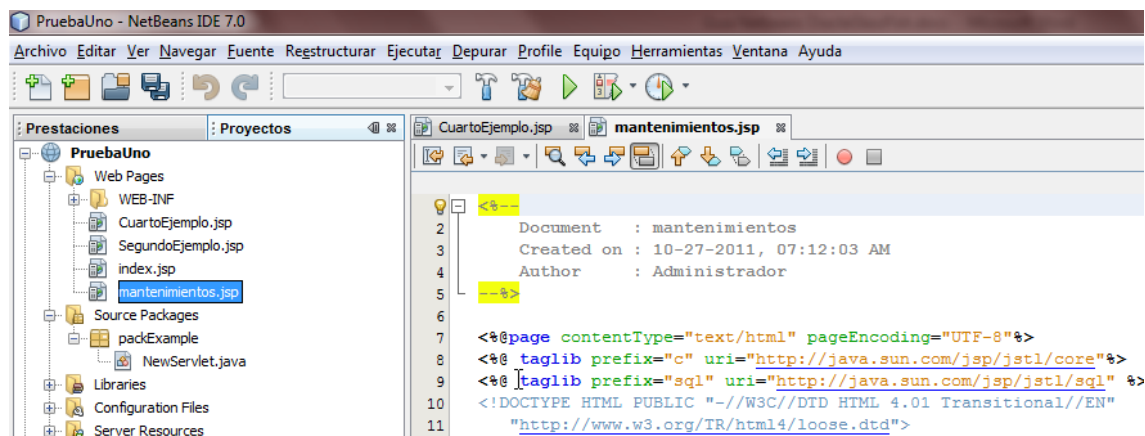
</TABLE>

</BODY>

</HTML>

```

Crear mantenimientos.jsp y reemplazar por el siguiente código



```
<%--
```

Document : mantenimientos

Created on : 10-27-2011, 07:12:03 AM

Author : Administrador

```
--%>
```

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

    "http://www.w3.org/TR/html4/loose.dtd">

<HTML>

<HEAD>

    <TITLE>Empleados Northwind </TITLE>

</HEAD>

<BODY>

<CENTER>

<H2>

    Listado de Empleados

</H2>

    <H4>

        Mantenimientos

    </H4>

</CENTER>

    <c:set var="opt" value="${param.opt}"/>

    <c:if test="${opt=='I'}">

        <form action="CuartoEjemplo.jsp" method="post">

```

<h4>Para crear nuevo empleado, favor completar formulario!</h4>

<dl>

<label for="">Nombre:</label> <input type="text"

name="FIRSTNAME" id="" size="54"/>

</dl>

<dl>

<label for="">Apellido:</label> <input type="text"

name="LASTNAME" id="" size="54"/>

</dl>

<dl>

<label for="">Cargo:</label> <input type="text" name="TITLE" id=""

size="54" />

</dl>

<input type="hidden" name="opt" value="I" />

<dl class="submit">

<input type="submit" name="Crear" id="submit" value="Crear" />

</dl>

</form>

</c:if>

<c:if test="{opt=='U'}">

<form action="CuartoEjemplo.jsp" method="post">

<h4>Para modificar nuevo empleado, favor completar
formulario!</h4>

<dl>

<label for="">Nombre:</label>

<label for=""><c:out value="\${param.FIRSTNAME}" /></label>

</dl>

<dl>

<label for="">Nuevo Nombre:</label>

<input type="text" name="FIRSTNAME" id="" size="54"/>

</dl>

<dl>

<label for="">Apellido:</label>

<label for=""><c:out value="\${param.LASTNAME}" /></label>

</dl>

<dl>

<label for="">Nuevo Apellido:</label>

<input type="text" name="LASTNAME" id="" size="54"/>

</dl>

<dl>

<label for="">Cargo:</label>

```

        <label for=""><c:out value="\${param.TITLE}" /></label>

    </dl>

    <dl>

        <label for="">Nuevo Cargo:</label>

        <input type="text" name="TITLE" id="" size="54" />

    </dl>

    <input type="hidden" name="opt" value="U" />

    <input type="hidden" name="EMPLOYEEID"
value="\${param.EMPLOYEEID}" />

    <dl class="submit">

        <input type="submit" name="Modificar" id="submit"
value="Modificar" />

    </dl>

</form>

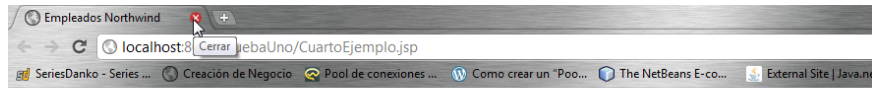
</c:if>

</BODY>

</HTML>

```

Ejecutar CuartoEjemplo.jsp y verificar resultados

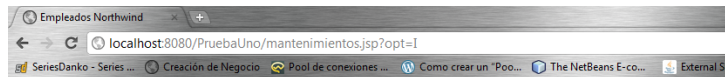


Listado de Empleados

Ejemplo 4 Insert, Update utilizando JNDI y JSTL

[Nuevo Empleado ++](#)

Codigo	Apellido	Nombre	Cargo	
1	uuu	uuu	uu	Modificar
2	Fuller	Andrew	Vice President, Sales	Modificar
3	Leverling	Janet	Sales Representative	Modificar
4	Peacock	Margaret	Sales Representative	Modificar



Listado de Empleados

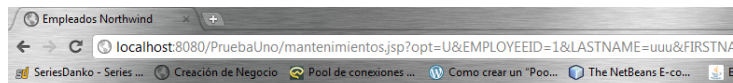
Mantenimientos

Para crear nuevo empleado, favor completar formulario!

Nombre:

Apellido:

Cargo:



Listado de Empleados

Mantenimientos

Para modificar nuevo empleado, favor completar formulario!

Nombre: uuu

Nuevo Nombre:

Apellido: uuu

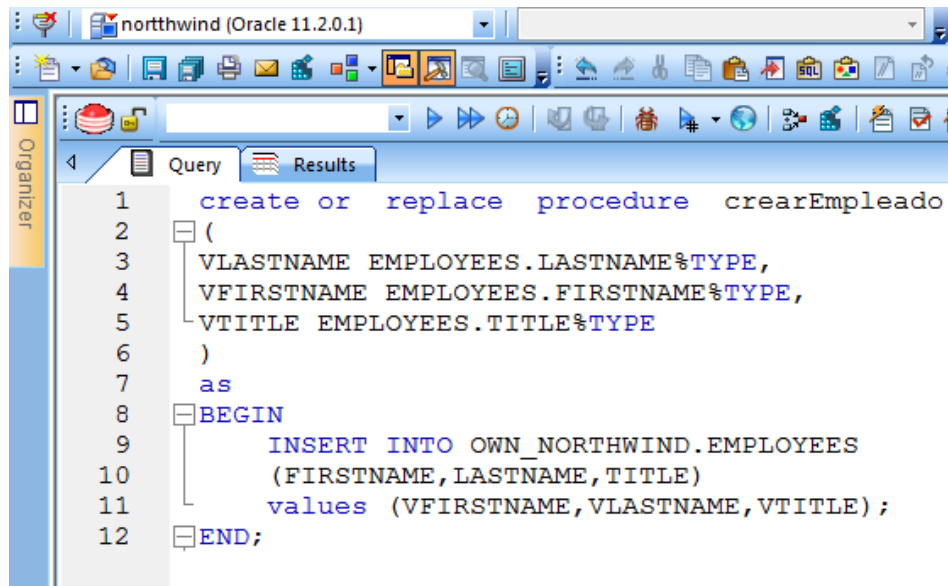
Nuevo Apellido:

Cargo: uu

Nuevo Cargo:

Ejemplo 5 Uso de Procedimientos almacenados desde Servlet's

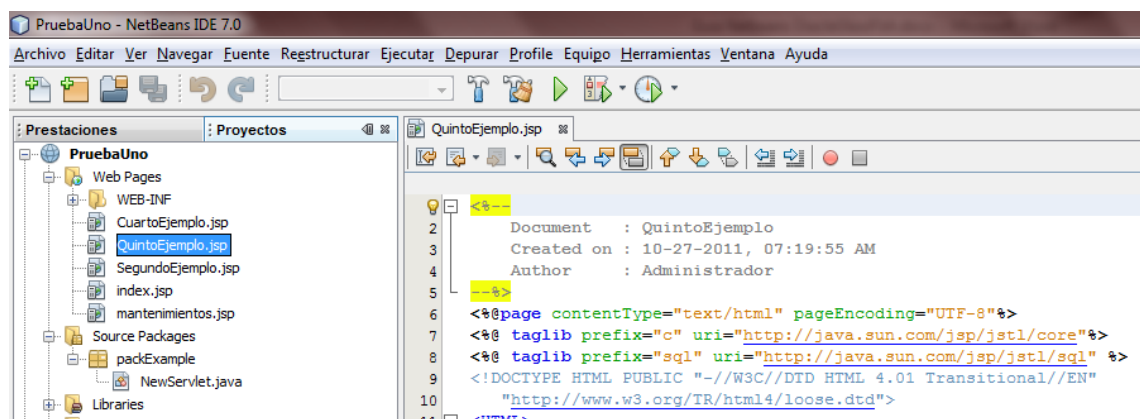
Crear procedimiento almacenado para insertar un registro en la tabla empleados



```
1 create or replace procedure crearEmpleado
2 (
3     VLASTNAME EMPLOYEES.LASTNAME%TYPE,
4     VFIRSTNAME EMPLOYEES.FIRSTNAME%TYPE,
5     VTITLE EMPLOYEES.TITLE%TYPE
6 )
7 as
8 BEGIN
9     INSERT INTO OWN_NORTHWIND.EMPLOYEES
10    (FIRSTNAME, LASTNAME, TITLE)
11    values (VFIRSTNAME, VLASTNAME, VTITLE) ;
12 END;
```

Crear nueva JSP con el nombre QuintoEjemplo.jsp

Copiar el código detallado a continuación:



```
Document : QuintoEjemplo
Created on : 10-27-2011, 07:19:55 AM
Author : Administrador

<@page contentType="text/html" pageEncoding="UTF-8">
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<HTML>
```

<%--

Document : QuintoEjemplo

Created on : 10-27-2011, 07:19:55 AM

Author : Administrador

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<HTML>

<HEAD>

<TITLE>Empleados Northwind </TITLE>

</HEAD>

<BODY>

<CENTER>

<H2>

Listado de Empleados

</H2>

<H4>

Ejemplo 5 Insert utilizando JNDI y Procedimientos almacenados desde un
servlet

```

</H4>

</CENTER>

<hr>

<sql:setDataSource var="conexion" dataSource="jdbc/myDatasource" />

    <a href="newEmpleado.jsp"><strong>Nuevo Empleado ++
</strong></a>

    <sql:query var="encabezado" dataSource="${conexion}">
SELECT  T.EMPLOYEEID, T.LASTNAME, T.FIRSTNAME, T.TITLE
FROM OWN_NORTHWIND.EMPLOYEES T order by T.EMPLOYEEID
</sql:query>

<TABLE id="rounded-corner" summary="Detalle de Empleados">

    <TR>

        <TH>Codigo</TH>

        <TH>Apellido </TH>

        <TH>Nombre</TH>

        <TH>Cargo</TH>

    </TR>

    <c:forEach var="fila1" items="${encabezado.rows}">

        <TR>

            <TD> <c:out value="${fila1.EMPLOYEEID}" /></TD>

            <TD> <c:out value="${fila1.LASTNAME}" /></TD>

            <TD> <c:out value="${fila1.FIRSTNAME}" /></TD>

```

```

<TD> <c:out value="\${fila1.TITLE}" /></TD>

</TR>

</c:forEach>

</TABLE>

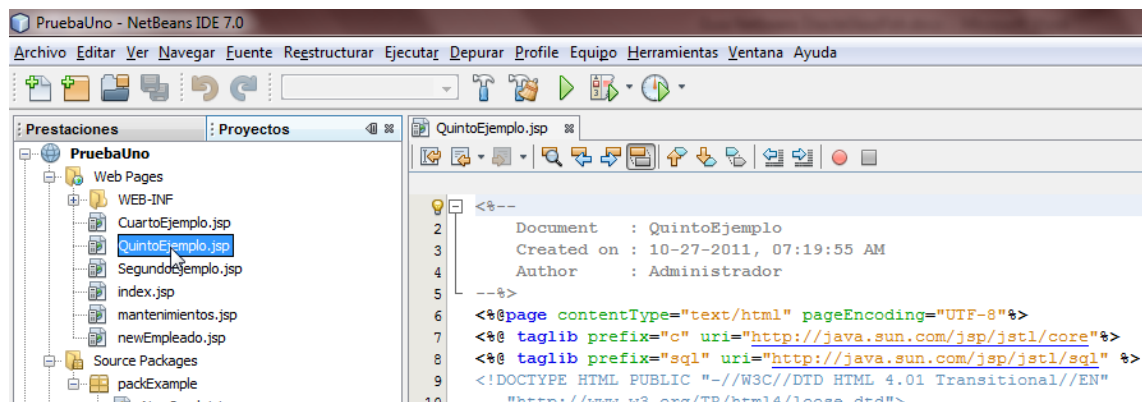
</BODY>

</HTML>

```

Crear nueva JSP con el nombre newEmpleado.jsp

Copiar el código detallado a continuación:



```
<%--
```

```
    Document : newEmpleado
```

```
    Created on : 10-27-2011, 07:23:50 AM
```

```
    Author : Administrador
```

```
--%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

    "http://www.w3.org/TR/html4/loose.dtd">

<HTML>

<HEAD>

    <TITLE>Empleados Northwind </TITLE>

</HEAD>

<BODY>

<CENTER>

<H2>

    Listado de Empleados

</H2>

    <H4>

        Mantenimientos

    </H4>

</CENTER>

    <form id='agregaEmpleado' name='agregaEmpleado'

action="/WebApplication2/servAgregarEmpleado" method='post'>

    <h4>Para crear nuevo empleado, favor completar formulario!</h4>
```

```

        <dl>

            <label for="">Nombre:</label> <input type="text"
name="FIRSTNAME" id="" size="54"/>

        </dl>

        <dl>

            <label for="">Apellido:</label> <input type="text"
name="LASTNAME" id="" size="54"/>

        </dl>

        <dl>

            <label for="">Cargo:</label> <input type="text" name="TITLE" id=""
size="54" />

        </dl>

        <dl class="submit">

            <input type="submit" name="Crear" id="submit" value="Crear" />

        </dl>

    </form>

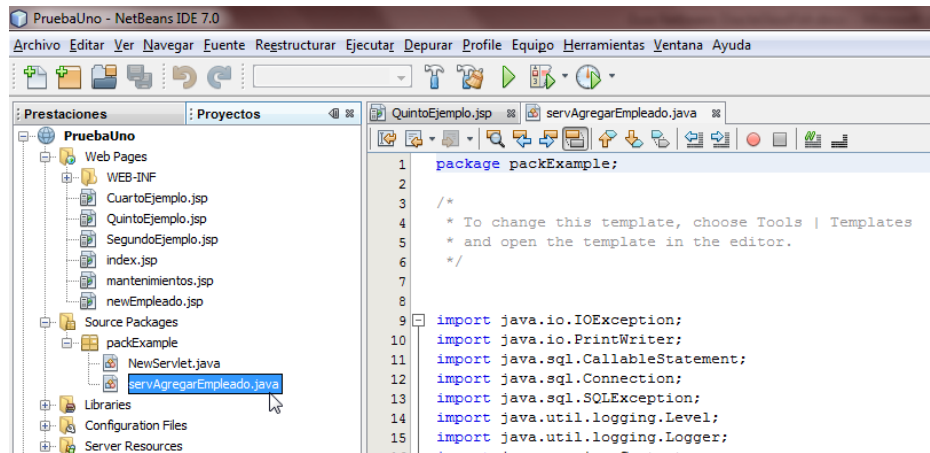
</BODY>

</HTML>

```

Crear nuevo servlet con el nombre servAgregarEmpleado

Copiar el código detallado a continuación:



```
package packExample;
```

```
/*
```

```
 * To change this template, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
import java.sql.CallableStatement;
```

```
import java.sql.Connection;
```

```
import java.sql.SQLException;
```

```

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.naming.NamingException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.sql.DataSource;

/**
 *
 * @author Administrador
 */
@WebServlet(name = "servAgregarEmpleado", urlPatterns =
{"/servAgregarEmpleado"})
public class servAgregarEmpleado extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)

```

```

        throws ServletException, IOException, NamingException, SQLException
    {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            Context ic = new InitialContext();

            DataSource dataSource = (DataSource)
ic.lookup("java:comp/env/jdbc/myDatasource");

            Connection connection = dataSource.getConnection();

            CallableStatement consulta=connection.prepareCall("{call
crearEmpleado(?,?,?)}");

            //lectura de parametros

            String pLASTNAME= request.getParameter("LASTNAME");
            String pFIRSTNAME = request.getParameter("FIRSTNAME");
            String pTITLE = request.getParameter("TITLE");

            //Asignar parametros al stored procedures

            consulta.setString("VFIRSTNAME", pFIRSTNAME);
            consulta.setString("VLASTNAME", pLASTNAME);
            consulta.setString("VTITLE", pTITLE);

            consulta.execute();

            connection.close();
        }
    }
}

```

```

        response.sendRedirect("QuintoEjemplo.jsp");
    } finally {
        out.close();
    }
}

```

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

```

/**
 * Handles the HTTP <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (NamingException ex) {

```

```
Logger.getLogger(servAgregarEmpleado.class.getName()).log(Level.SEVERE,  
null, ex);
```

```
    } catch (SQLException ex) {
```

```
Logger.getLogger(servAgregarEmpleado.class.getName()).log(Level.SEVERE,  
null, ex);
```

```
    }
```

```
}
```

```
/**
```

```
 * Handles the HTTP <code>POST</code> method.
```

```
 * @param request servlet request
```

```
 * @param response servlet response
```

```
 * @throws ServletException if a servlet-specific error occurs
```

```
 * @throws IOException if an I/O error occurs
```

```
 */
```

```
@Override
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse  
response)
```

```
    throws ServletException, IOException {
```

```
    try {
```

```

        processRequest(request, response);
    } catch (NamingException ex) {

Logger.getLogger(servAgregarEmpleado.class.getName()).log(Level.SEVERE,
null, ex);

        } catch (SQLException ex) {

Logger.getLogger(servAgregarEmpleado.class.getName()).log(Level.SEVERE,
null, ex);

        }
    }

/**
 * Returns a short description of the servlet.
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

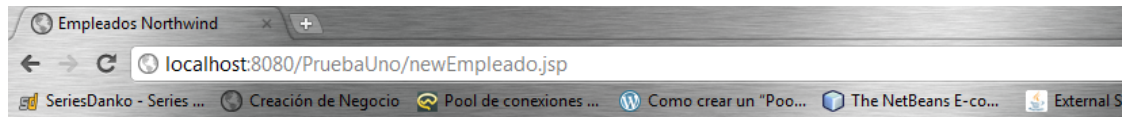
Ejecutar QuintoEjemplo.jsp y verificar resultados



Listado de Empleados

Ejemplo 5 Insert utilizando JNDI y Procedimientos almacenados desde un servlet

Nuevo Empleado ++			
Codigo	Apellido	Nombre	Cargo
1	uuu	uuu	uu
2	Fuller	Andrew	Vice President, Sales
3	Leverling	Janet	Sales Representative
...	...	...	...



Listado de Empleados

Mantenimientos

Para crear nuevo empleado, favor completar formulario!

Nombre:

Apellido:

Cargo: