

EXAMEN PRÁCTICO

1. Describe los pasos para realizar las siguientes acciones en Jdeveloper:
a) Establecer una interfaz gráfica menu.java con el título "Examen de Febrero" que contenga una barra de menú, en el que una de las opciones sea información acerca de la aplicación, y una barra de estado.

(0,5 puntos)

(Se puede explicar de distintas maneras)

- Desde el Navegador de Aplicaciones, hacemos **clic con el botón derecho** sobre la carpeta **View** de nuestra aplicación y elegimos **New**.
- Dentro de la categoría **Client Tier** ⇒ **Swing/AWT** elegimos el ítem **Frame** para crear un fichero fuente de tipo **Jframe**.
- En el cuadro de diálogo Create Frame, introducimos el título de la ventana y activamos las opciones Menu Bar, Status Bar y About Box, y le damos al botón Aceptar.

- b) Crear un Business Component de Tipo Entidad de la tabla Clientes que contenga los campos Idcliente, NombreEmpresa, Telefono, Email.

(0,5 puntos)

(Se puede explicar de distintas maneras)

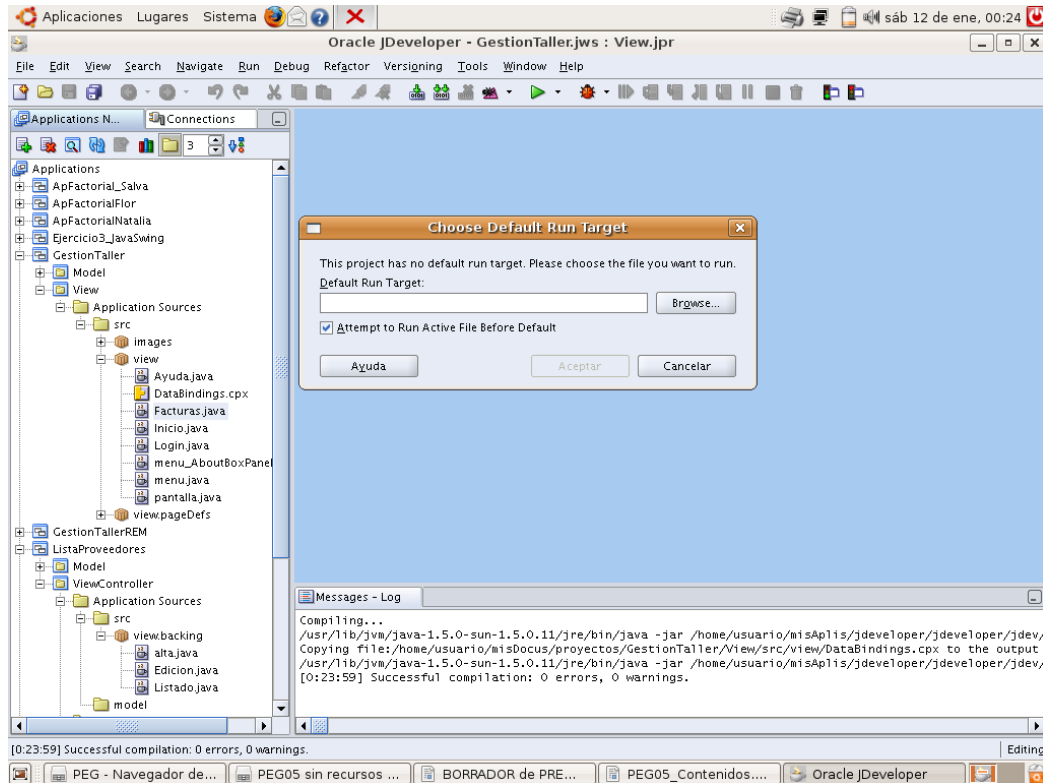
- Hacemos clic con el botón derecho sobre la carpeta **Model**
- Elegimos **New Entity Object**
- En uno de los pasos del asistente seleccionamos los campos Idcliente, NombreEmpresa, Telefono, Email

- c) Crear una conexión con una base de datos Oracle e introducir datos en las tablas. **(0,5 puntos)**

(Se puede explicar de distintas maneras)

- Debemos situarnos en nuestro proyecto abierto en JDeveloper en dirigirnos a la pestaña **Connection Navigator**, o bien en el menú **View/Connection Navigator**.
- Desde este panel haremos clic con el botón derecho en la carpeta **Database** que aparece colgada de **Connections**.
- Introducir los siguientes datos en el asistente: hostname (localhost), port (1521) y SID (XE)
- Para introducir los datos nos dirigimos a la pestaña **Data** de cada tabla e introducimos los registros correspondientes respetando las reglas de integridad

2.¿A qué es debido este mensaje en la ejecución de una aplicación? **(0,5 puntos)**
¿Cómo lo solucionarías? **(0,5 puntos)**. Las respuestas deben ser debidamente justificadas.



A que estamos ejecutando un proyecto en el que no está definido por defecto el origen de ejecución. Que no está main enetc. (Se puede explicar de distintas maneras)
Dando el origen correcto. O definiendo el main....etc (Se puede explicar de distintas maneras)

3.Como sabes, existen dos formas de trabajar con las bases de datos, o directamente con comandos SQL o a través de los ResultSet. ¿En tu opinión, qué ventajas aporta cada método? **(0.25 ptos)**

Una de la explicaciones posibles(la que aparece en los contenidos de la unidad 4): **El uso directo de SQL es más eficiente, pero el trabajo con ResultSet puede ser o suele ser más flexible y a veces imprescindible.**

Indica un supuesto en el que trabajarías con una base de datos directamente con comandos SQL y otro en que lo harías a través de ResultSet justificando detalladamente el por qué de tu elección. **(0.50 ptos)**

Una de la explicaciones posibles

Con SQL: P.e. Aplicación cliente/servidor (ncapas) con acceso remoto a datos: La aplicación procede a una actualización masiva de datos. Se lanzan las sentencias que modifican los datos lanzándolas desde la capa cliente o la de lógica de negocio. Una vez lanzadas no es necesaria la conexión con el servidor de datos (backend). Las sentencias se ejecutan en el servidor de datos y la capa cliente o la de lógica de negocio se desentienden del proceso. (Igual si lanzamos una sentencia de ejecución de un procedimiento almacenado. Pasamos los parámetros al procedure del backend (desde las capas antes indicadas) y la aplicación se desentiende del proceso)).

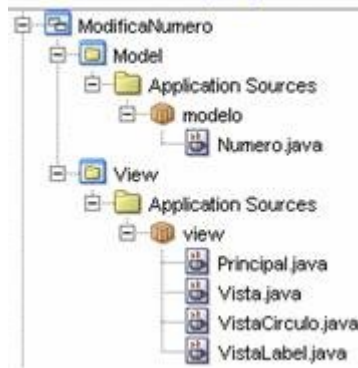
Así conseguimos no ocupar ancho de banda de la conexión en lo que dura el proceso. "Aseguramos" la integridad de los datos al ser el servidor de aplicaciones el que se encarga del proceso de actualización (este proceso no depende del estado de la conexión(velocidad,fiabilidad..etc), sino "sólo" del servidor de datos. Ejemplo típico de uso: Una Aplicación Web pero en general, cualquier aplicación cliente/servidor con acceso remoto a datos en el cual procedamos a modificación o borrado masivo de datos.

Con Resulset: Siguiendo con Aplicación cliente/servidor (ncapas) con acceso remoto a datos: P.e. Muestra de datos y navegación por los mismos. Permitir la manipulación no masiva de los mismos (edición, modificación, inserción y borrado). En el resulset almacenamos los datos seleccionados desde la capa cliente o log. de negocio y se muestran en la capa cliente, no ocupando espacio de la conexión(sólo en la transmisión de los datos desde el origen de datos hasta el cliente). Se permite así la interacción de los mismos por parte del usuario independientemente del estado de la conexión (obviamente, al liberarnos de la conexión(los datos del resulset "están en el cliente") ganamos en velocidad, seguridad, fiabilidad..etc). A la hora de actualizar los datos en el origen con alguna de las operaciones antes mencionadas, se realiza una conexión puntual con el backend para proceder a la misma.

Ejemplo típico de uso: Una Aplicación Web pero en general, cualquier aplicación cliente/servidor con acceso remoto a datos en el cual procedamos a edición de los datos (sobretudo si es masiva) o a inserción, modificación, o borrado NO masivo de datos.

4. Supongamos una aplicación de escritorio que utiliza el modelo MVC. La aplicación la llamaremos ModificaNumero. El modelo va a venir representado por una clase que llamaremos Numero. El interface Vista implementará todas las vistas que queramos tener, y consta de un método actualizar que será el que se ejecutará cuando haya un cambio en el modelo. La estructura sería la siguiente:

Estructura del proyecto



La información, los datos de la aplicación, se podrán representar de muchas formas. Pero esos datos son los mismos para todas las vistas.

En el proyecto correspondiente a la Vista, encontramos el fichero fuente Principal.java, desde el que ejecutaremos la aplicación en JDeveloper. Aquí es donde se crea un formulario con un campo de entrada (jTextField1) y un botón (jButton1).

Sabiendo todo esto y suponiendo que se escriba cualquier número y se pulse en el botón, debes escribir encima de cada línea del siguiente código la acción correspondiente que se ejecuta. **(0.75 ptos)**

```

private void jButton1_actionPerformed(ActionEvent e) {
    String texto = jTextField1.getText() ;
    Numero.setValor(texto) ;
    String valor = Numero.getValor() ;
    VistaCirculo.actualizar(valor);
    VistaLabel.actualizar(valor);
}

```

// Evento que se dispara cuando se pincha en el botón

```
private void jButton1_actionPerformed(ActionEvent e) {
```

// Obtener el contenido del campo de texto

```
String texto = jTextField1.getText() ;
```

// Establecer el contenido del campo de texto como el valor en el modelo

```
Numero.setValor(texto) ;
```

// Acceder al modelo para obtener los valores

```
String valor = Numero.getValor() ;
```

// Actualizar las vistas

```
vL.actualizar(valor);
```

```
vC.actualizar(valor);
```

```
}
```