|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Escuela Politécnica Superior  Ingeniería Informática  Prácticas de Sistemas Informáticos 2 | | | |
| **Grupo** | **2402** | **Práctica** | 1B | **Fecha** | 08/03/2021 |
| **Alumno/a** | | Rivas Molina, Lucía | | | |
| **Alumno/a** | | Santo-Tomás López, Daniel | | | |

# Práctica 1B: Arquitectura de JAVA EE

## Cuestión número 1:

**Abrir el archivo VisaDAOLocal.java y comprobar la definición de dicha interfaz. Anote en la memoria comentarios sobre las librerías Java EE importadas y las anotaciones utilizadas. ¿Para qué se utilizan? Comparar esta interfaz con el fichero de configuración del web service implementado en la práctica P1A.**

Están importadas las librerías java.sql para el acceso a la base de datos:

* java.sql.connection para establecer la conexión con la base de datos.
* Java.sql.resultset para almacenar los resultados de las consultas.
* Java.sql.sqlexception con las excepciones.
* Java.sql.statement para ejecutar sentencias.
* Javax.ejb.local para indicar que es local.

## Ejercicio 1:

**Introduzca las siguientes modificaciones en el bean VisaDAOBean para convertirlo en un EJB de sesión stateless con interfaz local:**

** Hacer que la clase implemente la interfaz local y convertirla en un EJB stateless mediante la anotación Stateless**

** Eliminar el constructor por defecto de la clase.**

** Ajustar el método getPagos() a la interfaz definida en VisaDAOLocal**

** Incluye en la memoria cada fragmento de código donde se han ido añadiendo las modificaciones solicitadas.**

Hemos añadido @stateless en vez del webservice en la clase VisaDaoBean.java y en el método getPagos() hemos vuelto a devolver arrays en vez de arraylists.

## Ejercicio 2:

**Modificar el servlet ProcesaPago para que acceda al EJB local. Para ello, modificar el archivo ProcesaPago.java de la siguiente manera: (…).**

**Importante: Esta operación deberá ser realizada para todos los servlets del proyecto que hagan uso del antiguo VisaDAOWS. Verifique también posibles errores de compilación y ajustes necesarios en el código al cambiar la interfaz del antiguo VisaDAOWS (en particular, el método getPagos()).**

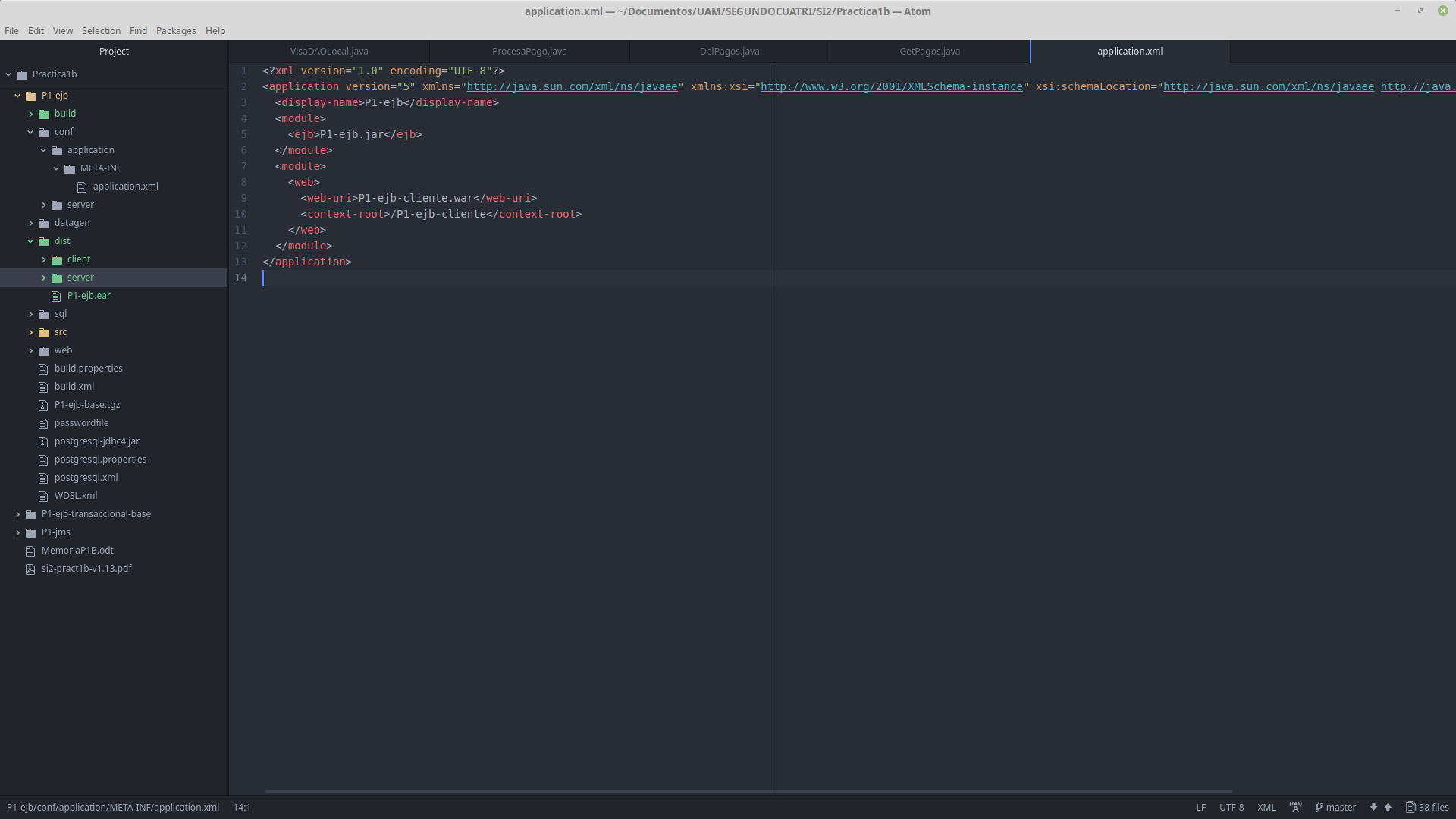
Todos los cambios se encuentran en el código que hemos modificado de la práctica. Como resumen, hemos seguido los pasos del enunciado: modificamos el servlet ProcesaPago.java añadiendo y eliminando unos imports, añadimos el objeto proxy al EJB local, eliminamos el webservice de la práctica anterior que estaba en un try catch y eliminamos las referencias a BindingProvider.

Además, hicimos los mismos cambios en los otros servlets que los necesitan, DelPagos.java y GetPagos.java, el cual hemos adaptado para que devuelva un array de PagoBean.

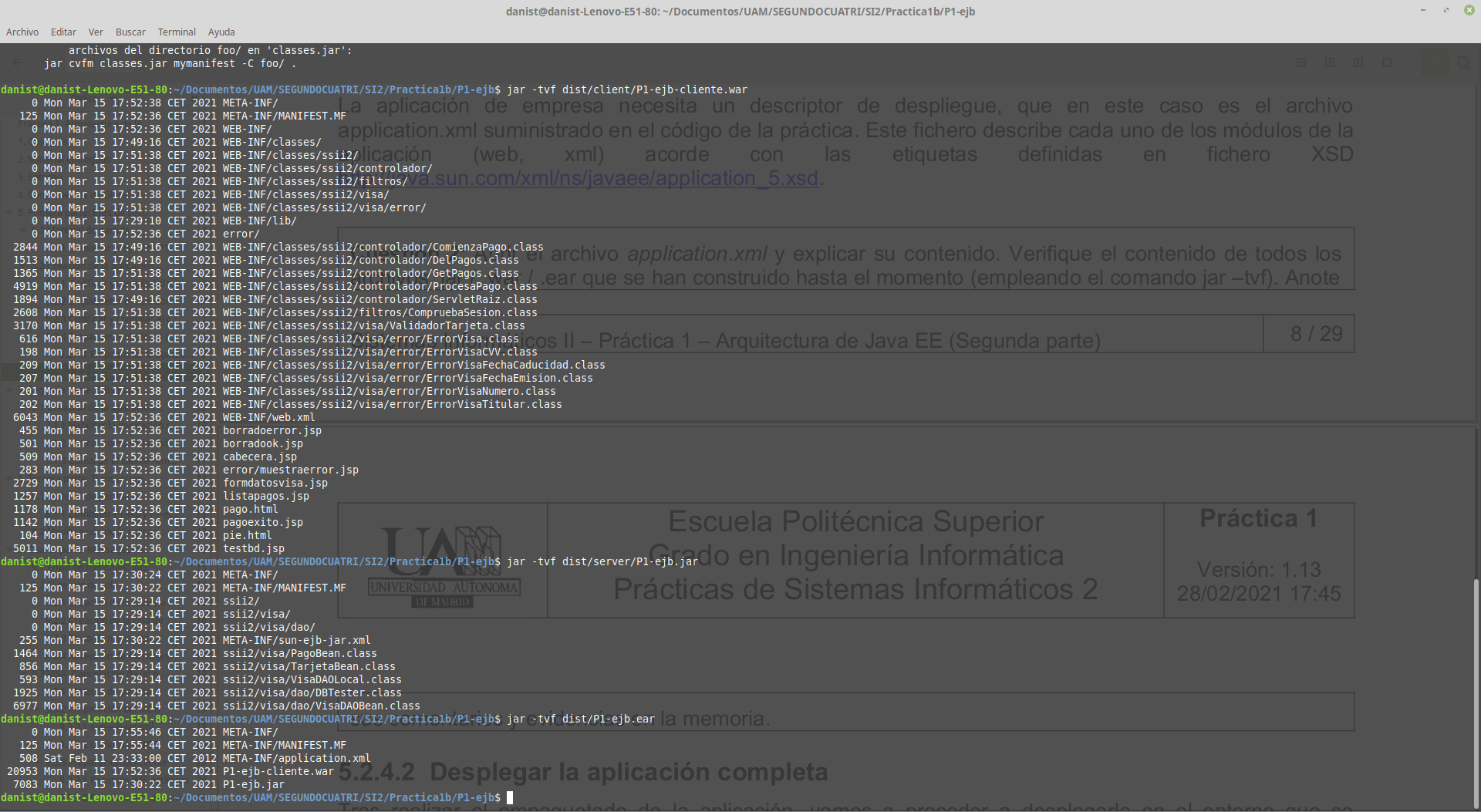
## **Cuestión número 2:**

**Editar el archivo application.xml y comprobar su contenido. Verifique el contenido de todos los archivos .jar /.war /.ear que se han construido hasta el momento (empleando el comando jar –tvf). Anote sus comentarios en la memoria.**

El archivo application.xml se encuentra en el directorio conf/application/META-INF, el cual hace referencia a P1-ejb.jar y a P1-ejb-cliente.war



Por un lado, el fichero .war contiene las páginas dinámicas y html, y además las clases de los servlets. El fichero .jar se genera cuando empaquetamos el servidor y contiene las clases del mismo y el descriptor del EJB. Finalmente, el fichero .ear contiene los ficheros anteriores y la descripción del application.xml, generándose al empaquetar la aplicación. Se adjunta la captura de dichos ficheros:



***Ejercicio 3:***

**Preparar los PCs con el esquema descrito y realizar el despliegue de la aplicación:**

** Editar el archivo build.properties para que las propiedades as.host.client y as.host.server contengan la dirección IP del servidor de aplicaciones. Indica qué valores y porqué son esos valores.**

** Editar el archivo postgresql.properties para la propiedad db.client.host y db.host contengan las direcciones IP adecuadas para que el servidor de aplicaciones se conecte al postgresql, ambos estando en servidores diferentes. Indica qué valores y porqué son esos valores.**

**Desplegar la aplicación de empresa**

Ahora las IPS son 10.2.1.2 para el cliente y servidor y 10.2.1.1 para la base de datos, luego modificamos el build.properties y postgre.sql como primer paso:

* as.host.client = 10.2.1.2
* as.host.server = 10.2.1.2
* db.host = 10.2.1.1
* db.client.host = 10.2.1.2

Desplegamos la aplicación con “ant desplegar” correctamente y pasamos al ejercicio siguiente para demostrar que verdaderamente funciona.

***Ejercicio número 4:***

**Comprobar el correcto funcionamiento de la aplicación mediante llamadas directas a través de las páginas pago.html y testbd.jsp (sin directconnection). Realice un pago. Lístelo. Elimínelo. Téngase en cuenta que la aplicación se habrá desplegado bajo la ruta /P1-ejbcliente.**

Primero realizamos un pago:

Luego comprobamos que se haya realizado accediendo a la lista de pagos:

Por último, eliminamos el pago:

***Ejercicio 5:***

**Realizar los cambios indicados en P1-ejb-servidor-remoto y preparar los PCs con el esquema de máquinas virtuales indicado. Compilar, empaquetar y desplegar de nuevo la aplicación P1- ejb como servidor de EJB remotos de forma similar a la realizada en el Ejercicio 3 con la Figura 2 como entorno de despliegue. Esta aplicación tendrá que desplegarse en la máquina virtual del PC2. Se recomienda replegar la aplicación anterior (EJB local) antes de desplegar ésta. Incluye en la memoria cada fragmento de código donde se han ido añadiendo las modificaciones solicitadas así como detallando los pasos realizados.**