CLASES	METODOS	EXCEPCIONES
FileWriter	write(valor);	IOException

BufferedWriter	write(valor);
	newLine();
		IOException

DataOutputStream	writeInt()
	writeDouble()
	writeUTF()
		IOException
		EOFException

ObjectOutputStream	writeObject()	FileNotFoundException IOException
FileReader	read();	IOException
BufferedReader	readLine();	IOException

DataIntputStream	readInt()
	readDouble()
	readUTF()
		EOFException
		IOException

ObjectInputStream	readObject()	EOFException FileNotFoundException IOException ClassNotFoundException
RandomAccessFile	void seek(long pos) long getFilePointer() long length() readBoolean(), readByte(), readInt(), readDouble(), readFloat() readUTF(), readLine() writeBoolean(), writeByte(), writeInt(), writeDouble(), writeFloat(), writeUTF(), writeLine()	EOFException FileNotFoundException IOException

SAX	SAXParserFactory spf=SAXParserFactory.newInstance();
	SAXParser sp=spf.newSAXParser();
	Manejador m=new Manejador();
	sp.parse();
		ParserConfigurationException
		SAXException
		IOException

XSTREAM	XStream xs = new XStream(new DomDriver());
	ObjectOutputStream oos=
	xs.createObjectOutputStream(new FileOutputStream());
	ObjectInputStream ois =
	
	xs.createObjectInputStream(new FileInputStream());
	PersistenceStrategy ps =
	new FilePersistenceStrategy(new File("."));
	XmlArrayList lista = new XmlArrayList(ps);
	xs.alias();
	xs.useAttributeFor();
	xstream.aliasField();
	String xml=xs.toXML();
	xs.fromXML(xml);
		IOException
		ClassNotFoundException
		FileNotFoundException

DOM	DocumentBuilderFactory dbf =
	DocumentBuilderFactory.newInstance();
	DocumentBuilder db = dbf.newDocumentBuilder();
	Document doc = db.newDocument();
	
	createElement(nombre)
	createAtribute(nombre)
	createTextNode(cadena)
	appendChild(nodo)
	replaceChild(nodo,referencia)
	getElementsByTagName()
	getAttribute(nombre)
	setAttribute(nombre,valor)
	getAttributeNode(nombre)
	setAttributeNode(attr)
	removeAttribute(nombre)
	
	text
	firstChild
	lastChild
	parentNode
	nodeName
	nodeType
	nodeValue
	attribute
	ownerDocument
	
	TransformerFactory tf = TransformerFactory.newInstance();
	Transformer transformer = tf.newTransformer();
	DOMSource ds= new DOMSource(doc);
	StreamResult sr= new StreamResult(new File());
	StreamResult sr1 = new StreamResult(System.out);
	transformer.transform();
		ParserConfigurationException
		TransformerException

```
import java.io.*;
public class MiClaseOutput extends ObjectOutputStream {
        MiClaseOutput(FileOutputStream f) throws IOException{
                super(f);
        }
   protected void writeStreamHeader() throws IOException{}
}
```

BASES DE DATOS

Conexion: mysql –h localhost –u root –p

Class.forName(“com.mysql.jdbc.Driver”);		ClassNotFoundException

Connection conn;
conn=DriverManager.getConnection(“jdbc:mysql://localhost/XX”,”root”,””);
XX nombre de la base de datos
		SQLException

Statement st;
st=conn.createStatement()
	execute()
	executeUpdate()
	executeQuery()
	close()
		SQLException

PreparedStatement pst;
pst = conn.prepareStatement("sentencia sql”);
	setInt(posicion,valor)
	setString(posicion,valor)
	setDouble(posicion,valor)
	close()
		SQLException

ResultSet rs;	next()
	getInt()
	getString()
	getDounble()
	close()
		SQLException

HIBERNATE

```
SessionFactory sf=HibernateUtil.sessionFactory();
Session sesion=sf.openSession();
Transaction t=sesion.beginTransaction();
```

```
public class HibernateUtil {
   static SessionFactory sessionFactory(){
    try {
         return new Configuration().configure().buildSessionFactory();
    } catch (Throwable ex) {
      System.err.println("Initial SessionFactory creation failed." + ex);
         throw new ExceptionInInitializerError(ex);
   }}}
```