

1. Heu d'implementar un servei SOAP que ens permeti consultar l'estoc de videojocs de la botiga.

//crear primer la interfície

```
package cat.xtec.ioc.service.soap;
import cat.xtec.ioc.domain.Videojoc;
import java.util.ArrayList;
import javax.jws.WebMethod;
import javax.jws.WebService;
import javax.jws.soap.SOAPBinding;
@WebService
@SOAPBinding(style = SOAPBinding.Style.DOCUMENT)
public interface VideojocServiceEndpoint {
    @WebMethod ArrayList<Videojoc> getAllGames();
    @WebMethod Videojoc getGame(String gameID);
    @WebMethod Videojoc getNumUnitatsVenudes(String gameID);
    @WebMethod Videojoc comprar(String gameID);
}
```

//implementació de la interfície

```
package cat.xtec.ioc.service.impl.soap;
import cat.xtec.ioc.domain.Videojoc;
import cat.xtec.ioc.repository.impl.InMemoryVideojocRepository;
import cat.xtec.ioc.service.soap.VideojocServiceEndpoint;
import java.util.ArrayList;
import javax.jws.WebService;

/**
 * @author jfaneca
 */
@WebService(serviceName = "VideojocService",
endpointInterface="cat.xtec.ioc.service.soap.VideojocServiceEndpoint ")
public class VideojocServiceEndpointImpl implements
VideojocServiceEndpoint{

    private final InMemoryVideojocRepository videojocRepository = new
InMemoryVideojocRepository();

    @Override
```

```

    public ArrayList<Videojoc> getAllGames() {
        return new
        ArrayList<Videojoc>(this.videojocRepository.getAllVideojocs());
    }

    @Override
    public Videojoc getGame(String gameID) {
        return this.getGame(gameID);
    }

    @Override
    public Videojoc getNumUnitatsVenudes(String gameID) {
        return this.getNumUnitatsVenudes(gameID);
    }

    @Override
    public Videojoc comprar(String gameID) {
        return this.comprar(gameID);
    }
}

```

2. Del webservice anterior, caldrà que enganxeu el codi del WSDL de contracte del servei.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.4.0 (wsit240-7e98ff4; 2017-08-03T21:19:54+0200) JAXWS-RI/2.3.0 JAXWS-API/2.3.0 JAXB-RI/2.3.0 JAXB-API/2.3.0 svn-revision#unknown. -->

<!-- Generated by JAX-WS RI (http://javaee.github.io/metro-jax-ws). RI's version is Metro/2.4.0 (wsit240-7e98ff4; 2017-08-03T21:19:54+0200) JAXWS-RI/2.3.0 JAXWS-API/2.3.0 JAXB-RI/2.3.0 JAXB-API/2.3.0 svn-revision#unknown. -->

<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://soap.impl.service.ioc.xtec.cat/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

```

```

xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://soap.impl.service.ioc.xtec.cat/" name="VideojocService">
    <import namespace="http://soap.service.ioc.xtec.cat/"
location="http://localhost:8080/dawm07eac5videojoc/VideojocService?wsdl=1"/>
    <binding xmlns:ns1="http://soap.service.ioc.xtec.cat/"
name="VideojocServiceEndpointImplPortBinding" type="ns1:VideojocServiceEndpoint">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"/>
        <operation name="getGame">
            <soap:operation soapAction=""/>
            <input>
                <soap:body use="literal"/>
            </input>
            <output>
                <soap:body use="literal"/>
            </output>
        </operation>
        <operation name="comprar">
            <soap:operation soapAction=""/>
            <input>
                <soap:body use="literal"/>
            </input>
            <output>
                <soap:body use="literal"/>
            </output>
        </operation>
        <operation name="getAllGames">
            <soap:operation soapAction=""/>
            <input>
                <soap:body use="literal"/>
            </input>
            <output>
                <soap:body use="literal"/>
            </output>
        </operation>
    </binding>
</wsdl:binding>
</wsdl:service>

```

```
        </output>
    </operation>
    <operation name="getNumUnitatsVenudes">
        <soap:operation soapAction=""/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
</binding>
<service name="VideojocService">
    <port name="VideojocServiceEndpointImplPort"
binding="tns:VideojocServiceEndpointImplPortBinding">
        <soap:address location="http://localhost:8080/dawm07eac5videojoc-
1.0- SNAPSHOT/VideojocService"/>
    </port>
</service>
</definitions>
```

3. Prepareu un servei REST en una classe que ens permeti consultar la llista de videojocs d'una botiga

```
package cat.ioc.xtec.service.rest;
import javax.ejb.Stateless;
import javax.ws.rs.*;
import javax.ws.rs.core.MediaType;
import java.util.List;
import cat.xtec.ioc.domain.Videojoc;
import cat.xtec.ioc.repository.impl.InMemoryVideojocRestRepository;

@Path("/videojoc")
@Stateless
public class VidejocRestService {

    private final InMemoryVideojocRestRepository
inMemoryVideojocRestRepository = new InMemoryVideojocRestRepository();

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public List<Videojoc> getAll() {
        return inMemoryVideojocRestRepository.getAllVideojocs();
    }

    @GET
    @Path("/{nom}")
    @Produces(MediaType.APPLICATION_JSON)
    public Videojoc findByName(@PathParam("nom") String nom){
        return inMemoryVideojocRestRepository.getVideojocByName(nom);
    }

    @GET
    @Path("getVideojocsVenuts/{nom}")
    @Produces(MediaType.APPLICATION_JSON)
    public Videojoc getEntradesVenudes(@PathParam("nom") String nom) {
        return inMemoryVideojocRestRepository.getVideojocVenut(nom);
    }

    @PUT
    @Path("/{nom}")
    @Consumes(MediaType.APPLICATION_JSON)
    public void comprarVideojoc(@PathParam("nom") String nom) {
        this.inMemoryVideojocRestRepository.comprarVideojoc(nom);
    }

    @POST
```

```
@Consumes(MediaType.APPLICATION_JSON)
public void afegirVideojoc(Videojoc videojoc) {
    this.inMemoryVideojocRestRepository.addVideojoc(videojoc);
}
}
```

4. Què indica la anotació @Produces de JAX-RS?

Què indica la anotació @Produces de JAX-RS?

Trieu-ne una:

- ☒ a. Respon a les peticions en les que la capçalera sigui del tipus MIME indicat. ✓
- ☐ b. L'operació CRUD que retorni el recurs.
- ☐ c. El tipus JAVA del resultat.
- ☐ d. L'estructura que genera el mètode.

5. Digueu si és certa la següent afirmació:

REST es basa a definir accions sobre recursos mitjançant l'ús dels mètodes GET, POST, PUT i DELETE inherents d'HTTP

Trieu-ne una:

- ☒ Vertader ✓
- ☐ Fals

6. Escriviu un possible client en Java per a un servei REST publicat a la URL <http://localhost:8080/consultaJocs/rest/>

Amb quina API trebal·leu? Treballarem amb l'API Client de JAX-RS.

```

package cat.xtec.ioc.consultajocs.client;

import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Invocation;
import javax.ws.rs.client.WebTarget;
import javax.ws.rs.core.GenericType;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.UriBuilder;
import java.net.URI;
import java.util.List;

public class ClientJocs {
    private static final Client NEW_CLIENT = ClientBuilder.newClient();

    public static void main(String[] args) {
        //preparació
        URI uri =
UriBuilder.fromUri("http://localhost:8080/consultaJocs/rest/jocs").port(8080)
        .build();
        WebTarget target = NEW_CLIENT.target(uri);
        Invocation invocation =
target.request(MediaType.APPLICATION_JSON).buildGet();
        //execució
        Response res = invocation.invoke();
        List<Jocs> returnedJocList = res.readEntity(new
GenericType<List<Jocs>>() {});
        //verificació status de la resposta
        System.out.println("GetStatus of response: " + res.getStatus());

        //iterar la llista
        for(Jocs joc : returnedJocList) {
            System.out.println(joc.toString());
        }
    }
}

```

7. Indiqueu si la següent afirmació és certa o falsa:

Un recurs REST només es pot accedir amb els verbs GET o PUT

Trieu-ne una:

☐ Vertader

☒ Fals ✓

8. Digueu si la següent afirmació és certa o falsa:

Una de les principals motivacions per fer un servei web RESTful asíncron és l'escalabilitat

Trieu-ne una:

- ☒ Vertader ✓
- ☐ Fals