

**CFGS Desenvolupament d'Aplicacions Web**  
**Mòdul 7 – Desenvolupament web en entorn servidor**  
**UF1 – Desenvolupament web en entorn servidor**  
**Unitat 2 – Desenvolupament web en entorn servidor**  
**EAC2**  
**(Curs 2022–23 / 1r semestre)**

**Presentació i resultats d'aprenentatge**

Aquest exercici d'avaluació continuada (EAC) es correspon amb els continguts treballats a la unitat 2 **“Desenvolupament web en entorn servidor”**

Els **resultats d'aprenentatge** que es plantegen són:


- Selecciona les arquitectures i tecnologies de programació web en entorn servidor, analitzant les seves capacitats i característiques pròpies
- Escriu sentències executables per un servidor web reconeixent i aplicant procediments d'integració del codi en llenguatges de marques.
- Escriu blocs de sentències embeguts en llenguatges de marques, seleccionant i utilitzant les estructures de programació.
- Desenvolupa aplicacions web embegudes en llenguatges de marques analitzant i incorporant funcionalitats segons especificacions.

**Criteris d'avaluació**

La puntuació màxima assignada a cada activitat s'indica a l'enunciat.

Els criteris que es tindran en compte per avaluar el treball de l'alumnat són els següents:

- Caracteritza i diferencia els models d'execució de codi al servidor i al client web.
- Reconeix els avantatges que proporciona la generació dinàmica de pàgines web i les seves diferències amb la inclusió de sentències de guions a l'interior de les pàgines web.
- Identifica els mecanismes d'execució de codi en els servidors web.
- Reconeix les funcionalitats que aporten els servidors d'aplicacions i la seva integració amb els servidors web.
- Identifica i caracteritza els principals llenguatges i tecnologies relacionats amb la programació web en entorn servidor.
- Verifica els mecanismes d'integració dels llenguatges de marques amb els llenguatges de programació en entorn servidor.
- Reconeix i avalua les eines de programació en entorn servidor.
- Identifica els mecanismes de generació de pàgines web a partir de llenguatges de marques amb codi encastat.
- Identifica les principals tecnologies associades.
- Utilitza etiquetes per a la inclusió de codi en el llenguatge de marques.
- Reconeix la sintaxi del llenguatge de programació que s'ha d'utilitzar.
- Escriu sentències simples i comprova els seus efectes en el document resultant.
- Utilitza directives per modificar el comportament predeterminat.

	Codi: I71	Exercici d'avaluació continuada 2	Pàgina 1 de 14
	Versió: 03	DAW_M07_EAC2_Enunciat_2223S1	Lliurament: 02/11/2022

## Formació professional

- Empra els diferents tipus de variables i operadors disponibles en el llenguatge.
- Identifica els àmbits d'utilització de les variables.
- Utilitza mecanismes de decisió en la creació de blocs de sentències.
- Fa servir i verifica el seu funcionament.
- Empra arrays per emmagatzemar i recuperar conjunts de dades.
- Crea i utilitza funcions.
- Usa formularis web per interactuar amb l'usuari del navegador web.
- Empra mètodes per recuperar la informació introduïda en el formulari.
- Afegeix comentaris al codi.
- Identifica els mecanismes disponibles per al manteniment de la informació que fa a un client web concret i assenyal·la els seus avantatges.
- Empra sessions per mantenir l'estat de les aplicacions web.
- Utilitza galetes per emmagatzemar informació en el client web i per recuperar el seu contingut.
- Identifica i caracteritza els mecanismes disponibles per a l'autenticació d'usuaris.
- Escriu aplicacions que integrin mecanismes d'autenticació d'usuaris.
- Fa adaptacions a aplicacions web existents com a gestors de continguts o altres.
- Utilitza eines i entorns per facilitar la programació, la prova i la depuració del codi.

### Forma i data de lliurament

Un cop finalitzat l'exercici d'avaluació continuada heu d'enviar el document des de l'apartat **Lliurament EAC2** de l'aula, dins del termini establert. Tingueu en compte que el sistema no permetrà fer lliuraments després de la data i hora indicades.


### Important

Heu d'entregar la **carpeta completa del projecte de Netbeans**, conjuntament amb aquest document amb les respostes degudament emplenades, i comprimir-ho tot amb un **compressor lliure (zip, 7z o tar.gz)**. RAR no és un compressor lliure i no es pot fer servir per lliurar fitxers.

El nom del fitxer serà el següent: **DAWM07\_EAC2\_Cognom1\_Inicial del cognom2.zip**. Els cognoms s'escriuran sense accents. Per exemple, l'estudiant *Joan García Santos* posaria el següent nom al seu fitxer de l'EAC2: **DAWM07\_EAC2\_Garcia\_S.zip**.

Substituïu **Nom i cognoms** de la capçalera per les vostres dades personals.

El termini de lliurament finalitza a les **23:55 h** del dia **02/11/2022**. La proposta de solució de l'EAC i les qualificacions es publicaran el dia 09/11/2022.

	Codi: I71	Exercici d'avaluació continuada 2	Pàgina 2 de 14
	Versió: 03	DAW_M07_EAC2_Enunciat_2223S1	Lliurament: 02/11/2022

## Enunciat

En aquesta EAC2 continuem amb el projecte Hungry Verds, però afegirem funcionalitat i a més canviarem de tecnologies.

Heu de partir del projecte dawm07eac2RestaurantEnunciat que trobareu publicat a l'aula.

### Tecnologies al client

Quant a les tecnologies, a la banda del client, en lloc de fer servir Jsp (amb JSTL), canviarem a pàgines HTML5. En la vostra vida professional, això us permetrà separar del tot el client del servidor. I això és bo? Doncs, pot servir per tenir diversos projectes client que facin servir el mateix programa a la part servidora (un client navegador web, una app Android, IOS o Windows 10, ...)

Seguint a la banda del client, també hem canviat Bootstrap per Materialize (<http://materializecss.com/>) que proporciona el mateix que Bootstrap però s'apropa més als patrons de disseny actuals com Material Design (<https://material.google.com/>)

Finalment, veureu un seguit de fitxers Java Script on fem servir aquest llenguatge, combinat amb jQuery i crides AJAX. Si, la part del client cridarà a la part servidora (els nostre veritable objectiu) amb AJAX, evitant en la mesura del possible tornar a carregar la pàgina sencera, de manera que l'usuari pot tenir una millor experiència d'ús.


No us preocupeu per això, perquè tota la banda del client es dona feta, tant pel que fa als enllaços necessaris per fer servir Materialize i jQuery, com el propi codi.

### Tecnologies al servidor

A l'EAC1 l'aplicació estava escrita amb JSP i JSTL, però heu de pensar que internament el servidor d'aplicacions (Glassfish al nostre cas) creava un Servlet per gestionar les peticions i construir dinàmicament les pàgines.

A l'EAC2 enriqueim el nostre projecte fent servir:

- Servlets que desenvolupareu vosaltres
- Classes normals com les que heu fet servir en altres mòduls (POJO [https://es.wikipedia.org/wiki/Plain\\_Old\\_Java\\_Object](https://es.wikipedia.org/wiki/Plain_Old_Java_Object))
- Classes especials com les que utilitzareu per les validacions
- Enterprise Java Bean (EJB) per diversos objectius com, per exemple, mantenir l'estat durant la sessió

	Codi: I71	Exercici d'avaluació continuada 2	Pàgina 3 de 14
	Versió: 03	DAW_M07_EAC2_Enunciat_2223S1	Lliurament: 02/11/2022

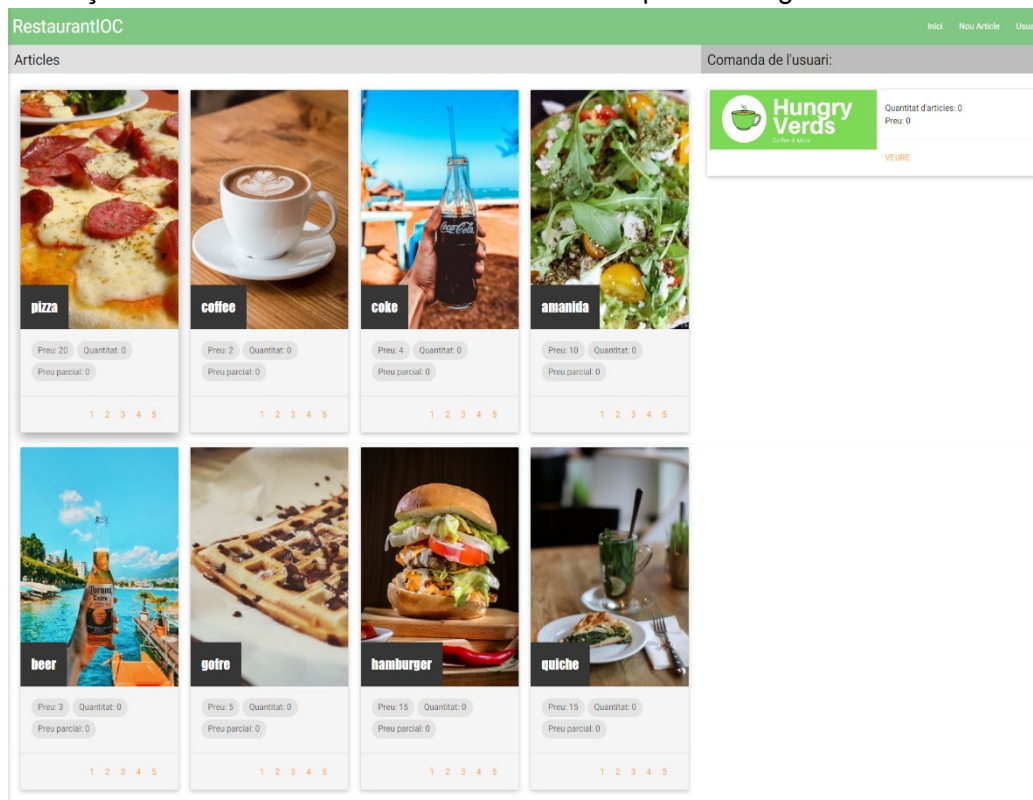
## Disseny de l'aplicació Restaurant a l'EAC2

Al final d'aquest enunciat, es donen els requeriments detallats de cada classe, propietat i mètode, però abans hem de descriure què ha de fer l'aplicació, com és i com ho ha de fer.

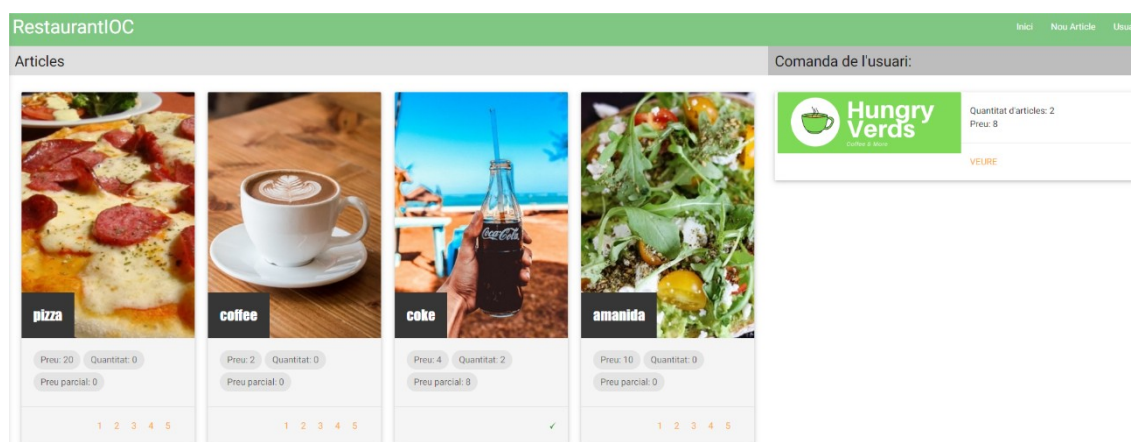
### Què ha de fer l'aplicació?

#### Pàgina inicial.

L'execució des del navegador, amb la url base (o index.html) ens ha de donar una llista dels articles que es poden comprar, més l'estat de la comanda (quantitat d'articles i preu total). Els enllaços 1 2 3 4 5 de cada article té com a conseqüència afegir-la a la comanda i es canvia



els enllaços pel tick.



## Formació professional

A la barra de menú hi ha 3 opcions. Tornar a Inici que mostrarà la pàgina inicial, Nou article per afegir... *wait for it...* **articles** que es podran comprar i Usuari per identificar-nos (encara que només serà per practicar la conservació d'estat durant la sessió)

### Nou Article

Si premeu Nou Article se'ns mostra el formulari per afegir un article a la carta.

RestaurantIOC

IniciNou ArticleUsuari

### Nou Article

Nom de l'article

Nom de l'article

Preu

Preu

IMATGE DE L'ARTICLE

La imatge ha de ser quadrada, entre 300x300 i 600x600 píxels.

El nom del fitxer ha de ser el nom del producte amb extensió jpg.

CREAR ➤

S'ha de poder entrar les dades, tenint en compte les restriccions que es donen, i a l'enviar, es tornarà a mostrar la pàgina inicial amb l'estat que tenia i l'article afegit.

RestaurantIOC

IniciNou ArticleUsuari

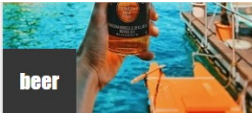
### Nou Article

Nom de l'article

lenties

Preu

12

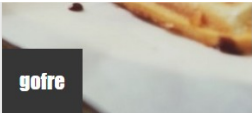


**beer**

Preu: 3   Quantitat: 0

Preu parcial: 0

1 2 3 4 5




**gofre**

Preu: 5   Quantitat: 0

Preu parcial: 0

1 2 3 4 5




**hamburger**

Preu: 15   Quantitat: 0

Preu parcial: 0

1 2 3 4 5




**quiche**

Preu: 15   Quantitat: 0

Preu parcial: 0

1 2 3 4 5




**lenties**

Preu: 12   Quantitat: 0

Preu parcial: 0

1 2 3 4 5



En el cas anterior, s'han complert totes les validacions. En canvi, si no s'omple algun camp, es fa la validació HTML5 al client, però si el que passa és que el nom de l'article no coincideix amb el de la imatge a carregar (sense extensió) llavors la validació es farà al servidor. En aquest darrer cas, es mostrarà la pàgina inicial sense l'article afegit (**això no és correcte perquè no informem a l'usuari, però ho fem per raons del que heu après fins aquest punt**)

### Usuari

Com en el cas de Nou Article, si premeu Usuari ens mostra el formulari per entrar les dades de l'usuari. En aquest cas, al entrar les dades i Desar, ens torna a la pàgina inicial, però si torneu a anar a Usuari, heu de veure que les dades continuen estant (mantenim doncs, l'estat)

RestaurantIOC

IniciNou articleUsuari

Nou usuari

Usuari

Usuari

Nom

Nom

Cognom

Cognom

DESAR ➤

RestaurantIOC

IniciNou articleUsuari

Nou usuari

Usuari

gflores

Nom


Germán

Cognom

Flores

DESAR ➤


Quan guardem l'usuari, aquest apareixerà a la pàgina principal (index.html).

	Codi: I71	Exercici d'avaluació continuada 2	Pàgina 6 de 14
	Versió: 03	DAW_M07_EAC2_Enunciat_2223S1	Lliurament: 02/11/2022

RestaurantIOC

IniciNou ArticleUsuari

ArticlesComanda de l'usuari: gflores



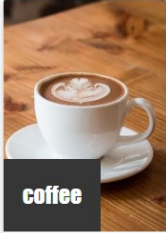
**pizza**

Preu: 20

Quantitat: 0

Preu parcial: 0

1 2 3 4 5



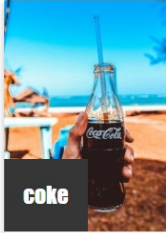
**coffee**

Preu: 2

Quantitat: 0

Preu parcial: 0

1 2 3 4 5




**coke**

Preu: 4

Quantitat: 2

Preu parcial: 8

✓




**amanida**

Preu: 10

Quantitat: 0

Preu parcial: 0

1 2 3 4 5



Quantitat d'articles: 1

Preu: 8

VEURE



## Comanda

En prémer VEURE a la pàgina inicial, ens ha de mostrar la pàgina amb la llista d'articles afegits a la comanda.

RestaurantIOC

Inici Nou Article Usuari

Articles

Comanda de l'usuari: gflores

**pizza**

Preu: 20 Quantitat: 3

Preu parcial: 60

**coffee**

Preu: 2 Quantitat: 0

Preu parcial: 0

**coke**

Preu: 4 Quantitat: 2

Preu parcial: 8

**amanida**

Preu: 10 Quantitat: 0

Preu parcial: 0

Quantitat d'articles: 2

Preu: 68

VEURE

RestaurantIOC

Inici Nou Article Usuari

Articles

Comanda

Article: coke Preu: 4 Quantitat: 2 Preu Parcial: 8

ELIMINAR

Article: pizza Preu: 20 Quantitat: 3 Preu Parcial: 60

ELIMINAR

Article: llenties Preu: 12 Quantitat: 3 Preu Parcial: 36

ELIMINAR

Quantitat articles: 3

Total: 104

TORNA

Si s'eliminen articles, s'ha de refrescar la llista i el resum.

RestaurantIOC

Inici Nou Article Usuari

Articles

Comanda

Article: pizza Preu: 20 Quantitat: 3 Preu Parcial: 60

ELIMINAR

Quantitat articles: 1

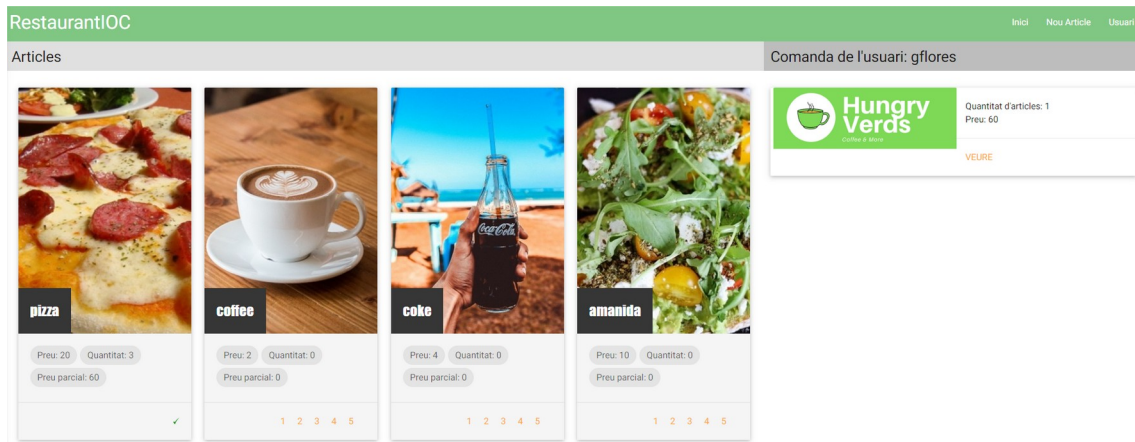
Total: 60

TORNA

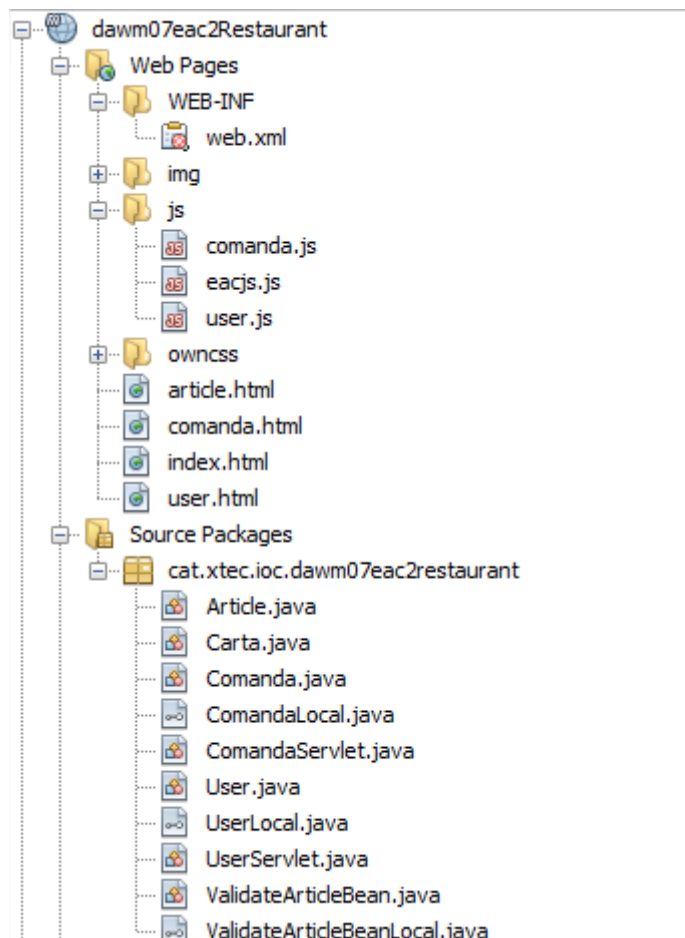



## Formació professional

Si ara es torna a la pàgina inicial, l'article eliminat ha quedat desmarcat i el resum actualitzat.



L'estructura de fitxers del projecte és la que es mostra a la imatge següent. Al llarg de l'enunciat de cada pregunta anirem veient cadascun d'ells i els desenvolupaments a fer.



	Codi: I71	Exercici d'avaluació continuada 2	Pàgina 9 de 14
	Versió: 03	DAW_M07_EAC2_Enunciat_2223S1	Lliurament: 02/11/2022

## Pregunta 1. Servlets. Desenvolupant la pàgina d'inici (3 punts)

Al iniciar l'aplicació des del navegador, amb la url base es fa una petició del fitxer *index.html*. Com aquest té associat *eac.js*, a banda d'execucions per al format, s'executarà la petició Ajax "Carta? action=listArticles"

Heu de desenvolupar la gestió d'aquesta petició:


- Al fitxer *web.xml* o per anotacions, heu de configurar que la petició l'agafi el servlet Carta del paquet *cat.xtec.ioc.dawm07eac2restaurant*
- El fitxer *web.xml* conté articles amb el seu preu. A *Carta.init* heu de carregar aquests articles sobre la llista d'articles propietat del Servlet. Cada article de la llista és de la classe *Article* que trobareu ja desenvolupada.
- A *Carta.ProcessRequest*, per no escriure tot el codi aquí, farem una bifurcació a altres mètodes en funció del paràmetre *action* (recordeu que la petició ara és "Carta? action=listArticles")
  - *action = listArticles*, bifurca al mètode **protected void listArticles(HttpServletRequest request, HttpServletResponse response)** que heu de desenvolupar en aquesta pregunta
  - *action = addArticleComanda*, bifurca al mètode **protected void addArticleComanda(HttpServletRequest request, HttpServletResponse response)** que desenvolupareu en aquesta pregunta
  - *action = createArticle*, bifurca al mètode **protected void createArticle(HttpServletRequest request, HttpServletResponse response)** que desenvolupareu en altres preguntes
- A *Carta.listArticles* heu de llegir els articles de la llista (articles) i retornar-los en un JSON. Concretament, seguiu aquestes instruccions al peu de la lletra:
  - Definiu les variables
    - *JSONObject json = new JSONObject();*
    - *JSONArray array = new JSONArray();*
  - Per a cada article de la llista
    - Definiu *LinkedHashMap<String, String> jsonOrderedMap = new LinkedHashMap<String, String>();*
    - Fiqueu cada camp a *jsonOrderedMap* on la key és el nom del camp
    - Fiqueu aquest *HashMap* dins d'un objecte *json*, exactament
      - *JSONObject member = new JSONObject(jsonOrderedMap);*
    - Afegiu aquest objecte a l'array JSON (*member* a *array*)
  - Al final afegiu l'array a l'objecte *json* inicial (la variable *json*) i imprimeu aquest
    - *json.put("jsonArray", array);*
    - *out.print(json.toString());*
    - //on *out* és el *PrintWriter*

Heu de tenir en compte que, avui en dia, la majoria d'aplicacions fan servir JSON com a protocol de dades entre client i servidor.

A la pàgina inicial, quan l'usuari prem una quantitat (número del 1 al 5), *eac.js* fa la petició AJAX "Carta?action=addArticleComanda&article=" + *article* + "&quantitat=" + *quantitat*;

Heu de desenvolupar la gestió d'aquesta petició.

- A *Carta.addArticleComanda*
  - Heu de recollir l'article i la quantitat que s'ha passat per paràmetre, buscar l'article de la llista d'articles que en coincideix i això us donarà l'objecte *article*

	Codi: I71	Exercici d'avaluació continuada 2	Pàgina 10 de 14
	Versió: 03	DAW_M07_EAC2_Enunciat_2223S1	Lliurament: 02/11/2022

corresponent, del qual heu d'afegir la quantitat i el subtotal (crida als mètodes **setQuantitat** i **setPreuparcial** de la classe Article)

○ A més, heu de retornar el nom de l'article afegit, el seu preu, la quantitat i el preuParcial en un JSON (aquesta és més simple) Heu de fer exactament:

- Definir un objecte JSON `JSONObject json = new JSONObject();`
- Afegir-li els valors amb keys "articleAfegit" pel nom de l'article, "preuArticle", "quantitatArticle" i "preuparcialArticle" per a la resta de valors. (instrucció `json.put(key,valor)`)
- Finalment, s'ha d'imprimir
  - `out.print(json.toString());`
  - `out.close();`
  - `//out és el PrintWriter`

### Pregunta 2. Formularis amb 'servlets' i EJB. Nou article. (3 punts)

A la pàgina inicial, quan l'usuari prem Nou Article, simplement es mostra la pàgina *article.html*. En aquesta pregunta, heu de desenvolupar els aspectes relacionats amb la creació d'un nou article, amb la càrrega d'imatges i validacions associades.


La pàgina *article.html* està associada al mateix fitxer Java Script *eac.js*, i per tant, hauríem d'atendre peticions AJAX des d'aquest fitxer. No obstant això, farem les peticions directament des de l'HTML. Si us fixeu a l'etiqueta form de *article.html*

```
<form method="POST" action="Article" id="fpform" name="fpform" enctype="multipart/form-data">
```

es tracta d'una petició POST on l'acció és Article i a més `enctype="multipart/form-data"` vol dir que enviarem un fitxer en la petició HTTP.

Heu de fer els següents desenvolupaments en l'aplicació.

- Al fitxer *web.xml* o per anotacions, heu de configurar que la petició (/Article) l'agafi també el servlet Carta del paquet `cat.xtec.ioc.dawm07eac2restaurant`
- Feu la validació consistent en que el nom de l'article enviat és igual al nom del fitxer d'imatge pujat sense l'extensió. L'heu de fer amb un EJB sense estat que implementi una interfície.
  - Interfície **ValidateArticleBeanLocal** (ha de ser local)
  - EJB **ValidateArticleBean** sense estat amb el mètode
    - **public Boolean isValidFileName(String articleName, String fileName)**
- A més a més, heu de comprovar que l'article no es trobi en el llistat (el nom de l'article ha de ser diferent). No poden haver-hi articles repetits.
- A *Carta.createArticle*
  - Heu de fer la validació (nom de l'article a donar d'alta igual al nom del fitxer i articles repetits)
  - Heu de pujar el fitxer que ens envien a la carpeta img del mateix projecte
    - Això no es pot fer directament. Heu de pujar el fitxer a la carpeta temporal que defineix l'anotació `@MultipartConfig(location ...)` i després copiar-lo a la carpeta img
      - **MOLT IMPORTANT:** si us dona un error pot ser perquè no existeixi la carpeta que poseu a la anotació en el directori de deploy del Glassfish. A la sortida del Glassfish server podreu veure el detall de a on s'hauria de crear la carpeta.
  - Heu de crear el nou article (classe Article) i afegir-la a la llista

	Codi: I71	Exercici d'avaluació continuada 2	Pàgina 11 de 14
	Versió: 03	DAW_M07_EAC2_Enunciat_2223S1	Lliurament: 02/11/2022

- Finalment heu de redirigir a *index.html* (això farà que es vegi el nou article a la llista de la pàgina inicial)

### Pregunta 3. Manteniment d'estat, autenticació i autorització amb Servlets i EJB. Usuari i Comanda. (4 punts)

En aquesta pregunta heu de desenvolupar els aspectes relatius al formulari Usuari i la gestió de les valoracions. En resum, es tracta de conservar l'estat durant la sessió.

#### Usuari

A la pàgina inicial, quan l'usuari prem Usuari, simplement es mostra la pàgina *usuari.html*. Aquesta pàgina està associada al fitxer Java Script *user.js* que enviarà les peticions AJAX a desenvolupar en la part servidor.

Heu de fer la gestió de la petició de mostrar el formulari (penseu que si ja s'havia omplert, ha de mostrar les dades de l'usuari actual). En concret, si observeu a *user.js*:

*"user?action=formUser"*

Heu de fer els següents desenvolupaments en l'aplicació.


- A la interfície **UserLocal** feu una validació tal que user no pugui tenir més de 10 caràcters. En cas contrari, el missatge ha de ser "Codi d'usuari no valid. Ha de ser menor de 10 caràcters."
- A **UserServlet** feu servir l'anotació que fa que aquest servlet agafi les peticions de *"user"*
- A *UserServlet.ProcessRequest*
  - Implementeu la derivació segons el paràmetre action
    - *action = formUser* crida al mètode **private void formUser(HttpServletRequest request, HttpServletResponse response)** (l'implementarem més tard)
    - *action = newUser* crida al mètode **private void newUser(HttpServletRequest request, HttpServletResponse response)**
- A *UserServlet.formUser*
  - Feu servir el bean UserLocal
    - Declareu UserLocal bean i agafeu el valor inicial d'un paràmetre de sessió "userbean"
    - Si el valor és null (no s'havia creat l'usuari abans), llavors heu de crear l'objecte bean i a més crear el paràmetre de sessió "userbean" amb aquest nou objecte bean
  - Què s'ha de retornar?
    - En qualsevol cas, un JSONObject amb els valors de l'usuari (amb keys els noms dels camps)
    - Si el bean s'acaba de crear, llavors els valors es retornen com "" (no torneu nulls)

També heu de fer la gestió de la petició de nou usuari. En concret, si observeu a *user.js*:

*"user?action=newUser&user=" + \$("#user").val()+"&name=" + \$("#name").val()+"&lastname=" + \$("#lastname").val();*

És a dir, ens envien totes les dades de l'usuari (no arriben en blanc perquè les validacions de camps buits es fan al client)

Heu de fer els següents desenvolupaments en l'aplicació.

	Codi: I71	Exercici d'avaluació continuada 2	Pàgina 12 de 14
	Versió: 03	DAW_M07_EAC2_Enunciat_2223S1	Lliurament: 02/11/2022

- A `UserService.newUser`
  - Feu servir el bean `UserLocal`
    - Recupereu l'objecte bean del paràmetre de sessió "userbean"
    - Actualitzeu les seves propietats amb els valors dels paràmetres de la request (user, name i lastname)
    - Feu la validació del nom
  - Què s'ha de retornar?
    - Si passa la validació, un objecte JSON simple amb key "resposta" i valor "OK"
    - Si no passa la validació, un objecte JSON simple amb key "resposta" i valor el missatge de la validació

### Comanda

A la pàgina inicial, quan l'usuari prem VEURE, simplement es mostra la pàgina *comanda.html*. Aquesta pàgina està associada al fitxer Java Script *comanda.js* que enviarà les peticions AJAX a desenvolupar en la part servidor.

Heu de fer la gestió de la petició de mostrar la pàgina. En concret, la petició "*comanda? action=articlesComandaList*" de *comanda.js*

Heu de fer els següents desenvolupaments en l'aplicació.

- Farem servir la interfície `ComandaLocal` i la implementació `Comanda` que contindrà la llista dels articles afegits. Recordeu que 'articles' de *Carta.java* conté la llista de tots els articles.
- A `ComandaServlet` feu servir l'anotació que fa que aquest servlet agafi les peticions de "/comanda"
- A `ComandaServlet.ProcessRequest`
  - Implementeu la derivació segons el paràmetre action
    - *action = articlesComandaList* crida al mètode **private void articlesList(HttpServletRequest request, HttpServletResponse response)** (l'implementarem més tard)
    - *action = deleteArticle* crida al mètode **private void deleteArticle(HttpServletRequest request, HttpServletResponse response)**
- A `ComandaServlet.articlesList`
  - Feu servir el bean `ComandaLocal`
    - Declareu *ComandaLocal comandabean* i agafeu el valor inicial d'un paràmetre de sessió "comandabean"
    - Si el valor és null, llavors heu de crear l'objecte *comandabean* i a més crear el paràmetre de sessió "comandabean" amb aquest nou objecte *comandabean*
    - Nota: inicialitzeu *articlesAfegits* amb un *new ArrayList*
  - Què s'ha de retornar?
    - En qualsevol cas, un *JSONObject* amb els articles del *comandabean* (*articlesAfegits*) però feu servir un *JSONArray* com a *Carta.listArticles* de la pregunta 1
    - Nota: Retorneu el nom (key = name), el preu (key=preu), la quantitat (key=quantitat) i el preuParcial (key=preuparcial) de tots els articles (les keys del JSON són els noms dels camps)

També heu de fer la gestió de la petició d'eliminar un article afegit. En concret, si observeu a `comanda.js`:

```
"comanda?action=deleteArticle&article=" + article;
```

Heu de fer els següents desenvolupaments en l'aplicació.

- A *ComandaServlet.deleteArticle*
  - Feu servir el bean *ComandaLocal*
    - Recupereu l'objecte *comandabean* del paràmetre de sessió "comandabean"
    - Elimineu l'article d'igual nom al passat per paràmetre de la llista *articlesAfegits* del *comandabean*, però abans elimineu l'última quantitat i preu parcial que es van afegir (mètodes **setQuantitat()** i **setPreuparcial()**).
  - Què s'ha de retornar?
    - Un objecte JSON simple amb key "resposta" i valor "OK"

I com afegim els articles al *comandabean*? Doncs heu de fer canvis a *Carta.java*.

- A *Carta.listArticles*
  - Feu una crida a **checkArticleAfegitSession(request, article)** que retornarà `true` o `false`
    - Si l'article que estem tractant per afegir al JSON de retorn està al *comandabean*, llavors heu d'afegir un element amb key "afegit" al Map amb valor "SI".
    - En cas contrari, amb valor "NO". Això farà, que surti el tick o els enllaços (1,2..5) per afegir a la llista d'articles de la pàgina inicial
- A *Carta.addArticleComanda*
  - Fer una crida **addArticleToSession(request, unArticle)**; que heu de desenvolupar al mateix servlet *Carta.java* ( un Article és l'article que esteu tractant de la llista d'articles)
- A *Carta.addArticleToSession*
  - Heu d'afegir l'article al bean de sessió *comandabean*.
- A *Carta.checkArticleAfegitSession*
  - Si el nom de l'article coincideix amb el nom d'algun article de la llista de *comandabean*, llavors es retorna `true`.
  - En cas contrari, es retorna `false`