

**CFGS Desenvolupament d'Aplicacions Web**  
**Mòdul 7 B2 – Desenvolupament web en entorn servidor**  
**UF4 - Serveis web. Pàgines dinàmiques interactives. Webs**  
**Híbrids**  
**Unitat 6 – Serveis web amb Spring**  
**EAC6**  
**(Curs 2022–23 / 2n semestre)**

**Presentació i resultats d'aprenentatge**

Aquest exercici d'avaluació continuada (EAC) es correspon amb els continguts treballats a la unitat 6 “Serveis web amb Java Spring ”

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


- Desenvolupa serveis web analitzant el seu funcionament i implantant l'estructura dels seus components.
- Genera pàgines web dinàmiques analitzant i utilitzant tecnologies del servidor web que afegixin codi al llenguatge de marques.
- Desenvolupa aplicacions web híbrids seleccionant i utilitzant llibreries de codi i dipòsits heterogenis d'informació.

**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:

- S'han identificat les característiques pròpies i l'àmbit d'aplicació dels serveis web.
- S'han identificat els avantatges d'utilitzar serveis web per a proporcionar accés a funcionalitats incorporades a la lògica de negoci d'una aplicació.
- S'han identificat les tecnologies i els protocols implicats en la publicació i utilització de serveis web.
- S'ha programat un servei web.
- S'ha creat el document de descripció del servei web.
- S'ha verificat el funcionament del servei web.
- S'ha consumit el servei web.
- S'han identificat les diferències entre l'execució de codi al servidor i al client web.
- S'han reconegut els avantatges d'unir les dues tecnologies en el procés de desenvolupament de programes.
- S'han identificat les llibreries i les tecnologies relacionades amb la generació per part del servidor de pàgines web amb guions embeguts.
- S'han utilitzat aquestes tecnologies per a generar pàgines web que incloguin interacció amb l'usuari en forma d'advertències i peticions de confirmació.
- S'han utilitzat aquestes tecnologies, per a generar pàgines web que incloguin verificació de formularis.
- S'han utilitzat aquestes tecnologies per a generar pàgines web que incloguin modificació dinàmica del seu contingut i la seva estructura.
- S'han aplicat aquestes tecnologies en la programació d'aplicacions web.

	Codi: I71	Exercici d'avaluació continuada 6	Pàgina de
	Versió: 01	DAW_M07B2_EAC6_Enunciat_2223S2	Lliurament: 16/05/2023

## Formació professional

- S'han reconegut els avantatges que proporciona la reutilització de codi i l'aprofitament d'informació ja existent.
- S'han identificat llibreries de codi i tecnologies aplicables en la creació d'aplicacions web híbrids.
- S'ha creat una aplicació web que recuperi i processi dipòsits d'informació ja existents.
- S'han creat dipòsits específics a partir d'informació existent a Internet i en magatzems d'informació.
- S'han utilitzat llibreries de codi per incorporar funcionalitats específiques a una aplicació web.
- S'han programat serveis i aplicacions web utilitzant com a base informació i codi generats per tercers.
- S'han provat, depurat i documentat les aplicacions generades.

### Forma i data de lliurament

Un cop finalitzat l'exercici d'avaluació continuada heu d'enviar el fitxer des de l'apartat **Lliurament EAC6** 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**, 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.

**Esborreu la carpeta *target/* del projecte abans de lliurar.**

El nom del fitxer serà el següent: **DAWM07B2\_EAC6\_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'EAC5: **DAWM07B2\_EAC6\_Garcia\_S.zip**.

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

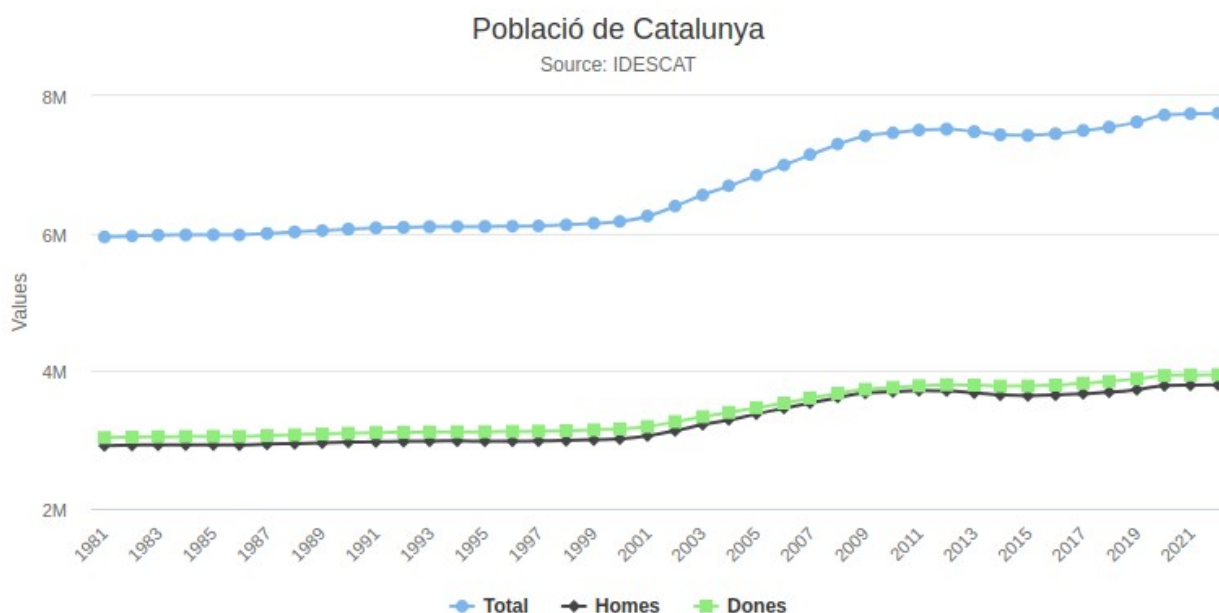
El termini de lliurament finalitza a les **23:55 h** del dia **16/05/2023**. La proposta de solució de l'EAC i les qualificacions es publicaran el dia 24/05/2023.

	Codi: I71	Exercici d'avaluació continuada 6	Pàgina de
	Versió: 01	DAW_M07B2_EAC6_Enunciat_2223S2	Lliurament: 16/05/2023

## Enunciat

### Població de Catalunya

Volem fer una aplicació per visualitzar la gràfica de com ha evolucionat la població de Catalunya els últims anys. La part del client la farem al final de l'EAC, i començarem centrant-nos en la part de servidor, que serà una aplicació JAVA que donarà un servei web RESTful sobre Spring.



Les dades que farem servir les descarregaràs del IDESCAT (Institut d'Estadística de Catalunya), de la següent url:

- <https://www.idescat.cat/indicadors/?id=anuals&n=10328&tema=xifpo>

Has de descarregar la sèrie temporal. Un cop obris el fitxer CSV es farà evident quines línies són les importants. Crea el fitxer *dades.csv* on posaràs les teves dades d'interès netes (sense capçalera, sense línies buides al final, ordenat per any):

```
1981;5949829;2916527;3033302
1982;5962723;2923258;3039465
1983;5974821;2928748;3046073
...
```

Hauràs vist que les columnes es corresponen a l'any, població total, homes i dones.

**NOTA:** En l'entrega, juntament amb la resta de fitxers del projecte, crearàs la carpeta ***captures/***, i adjuntaràs les captures que se't demanen en els diferents exercicis. En aquestes captures s'haurà de posar de manifest l'autoria.

## Pregunta 1 (2 punts). Començant el projecte

Se't proporciona un esquelet per importar a Netbeans, amb les dependències principals del projecte que necessitaràs. Amb el Glassfish arrencat, executa el projecte. Hauria de compilar sense errors i la única cosa que funciona és:

- <http://localhost:8080/poblaciocatalunya/index.jsp>



**Hello World!**

Veuràs que ja tenim creada la classe *Year.java*, que només conté la propietat *any/year (int)*. Completa aquesta classe amb les altres dades del domini (total, homes i dones, tots ells *int*).

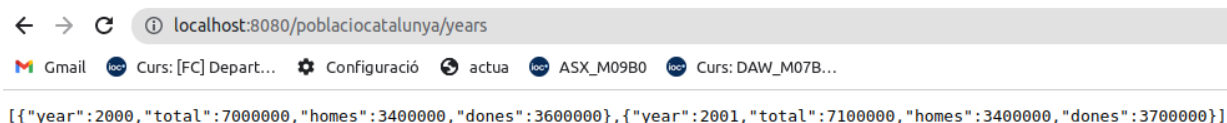
Completa les classes *YearRepository.java* i *InMemoryYearRepository.java*, de manera que carregaràs dos anys de prova:

```
Year first = new Year(2000, 7000000, 3400000, 3600000);
Year second = new Year(2001, 7100000, 3400000, 3700000);
```

Completa el controlador *YearsController.java* de manera que funcionin els endpoints */years* i */years/{year}* i puguis visualitzar les dades de prova.

```
@RequestMapping(method = RequestMethod.GET, value = "/years")
@RequestMapping(method = RequestMethod.GET, value = "/years/{year}")
```

- <http://localhost:8080/poblaciocatalunya/years>
- <http://localhost:8080/poblaciocatalunya/years/2000>




(l'alumne/a adjunta la captura de pantalla)

## Pregunta 2 (2 punts). Carregar les dades reals

Copiem la carpeta *dades/dades.csv* dins de *resources/*

L'objectiu és llegir de forma seqüencial el fitxer *dades.csv* i omplir una llista *List<Year>* amb tots els valors. Es pot fer de diverses maneres, però bàsicament és llegir el fitxer de forma seqüencial, i a mida que anem

	Codi: I71	Exercici d'avaluació continuada 6	Pàgina de
	Versió: 01	DAW_M07B2_EAC6_Enunciat_2223S2	Lliurament: 16/05/2023

## Formació professional

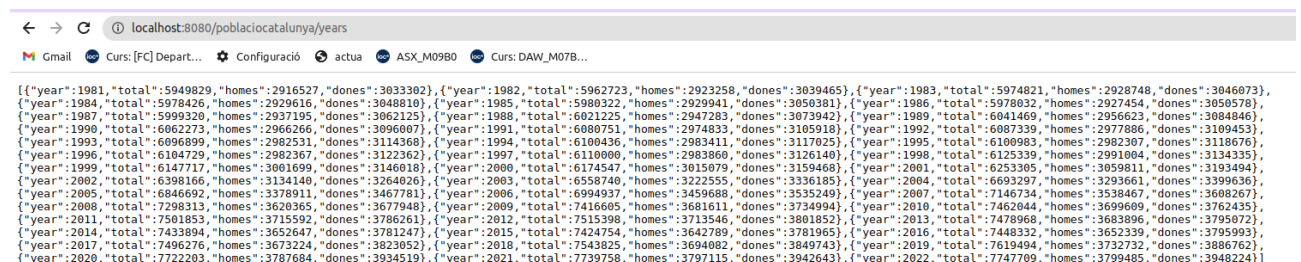
recuperant les files fer un *split* per separar els camps, creant un objecte *Year* per cada fila, i afegir-lo al repositori. S'ha de tenir en compte que quan llegim del fitxer tenim *Strings*, i els nostres valors són tots ells *int*.

Per ex:

```
...
Year year = new Year(Integer.parseInt(tempArr[0]),
Integer.parseInt(tempArr[1]), Integer.parseInt(tempArr[2]),
Integer.parseInt(tempArr[3]));
years.add(year);
...
```

Un cop et funcioni ja podràs visualitzar les dades reals:

- <http://localhost:8080/poblaciocatalunya/years>



```
{
  "year": 1981, "total": 5949829, "homes": 2916527, "dones": 3033302,
  "year": 1982, "total": 5962723, "homes": 2923258, "dones": 3039465,
  "year": 1983, "total": 5974821, "homes": 2928748, "dones": 3046073,
  "year": 1984, "total": 5978426, "homes": 2929616, "dones": 3048810,
  "year": 1985, "total": 5980322, "homes": 2929941, "dones": 3050381,
  "year": 1986, "total": 5978032, "homes": 2927454, "dones": 3050578,
  "year": 1987, "total": 5999320, "homes": 2937195, "dones": 3062125,
  "year": 1988, "total": 6021225, "homes": 2947283, "dones": 3073942,
  "year": 1989, "total": 6041469, "homes": 2956623, "dones": 3084846,
  "year": 1990, "total": 6062273, "homes": 2966266, "dones": 3096007,
  "year": 1991, "total": 6080751, "homes": 2974833, "dones": 3105918,
  "year": 1992, "total": 6087339, "homes": 2977886, "dones": 3109453,
  "year": 1993, "total": 6096899, "homes": 2982531, "dones": 3114368,
  "year": 1994, "total": 6100436, "homes": 2983411, "dones": 3117025,
  "year": 1995, "total": 6100983, "homes": 2982307, "dones": 3118676,
  "year": 1996, "total": 6104729, "homes": 2982367, "dones": 3122362,
  "year": 1997, "total": 6110000, "homes": 2983860, "dones": 3126140,
  "year": 1998, "total": 6125339, "homes": 2991004, "dones": 3134335,
  "year": 1999, "total": 6147717, "homes": 3001699, "dones": 3146018,
  "year": 2000, "total": 6174547, "homes": 3015079, "dones": 3159468,
  "year": 2001, "total": 6253305, "homes": 3059611, "dones": 3193494,
  "year": 2002, "total": 6398166, "homes": 3134140, "dones": 3264026,
  "year": 2003, "total": 6558740, "homes": 3222555, "dones": 3336185,
  "year": 2004, "total": 6693297, "homes": 3293661, "dones": 3399636,
  "year": 2005, "total": 6846692, "homes": 3378911, "dones": 3467781,
  "year": 2006, "total": 6994937, "homes": 3459688, "dones": 3535249,
  "year": 2007, "total": 7146734, "homes": 3538467, "dones": 3608267,
  "year": 2008, "total": 7298313, "homes": 3620365, "dones": 3677948,
  "year": 2009, "total": 7416605, "homes": 3681611, "dones": 3734994,
  "year": 2010, "total": 7462044, "homes": 3699609, "dones": 3762435,
  "year": 2011, "total": 7501853, "homes": 3715592, "dones": 3786261,
  "year": 2012, "total": 7515398, "homes": 3713546, "dones": 3801852,
  "year": 2013, "total": 7478968, "homes": 3683896, "dones": 3795072,
  "year": 2014, "total": 7433894, "homes": 3652647, "dones": 3781247,
  "year": 2015, "total": 7424754, "homes": 3642789, "dones": 3781965,
  "year": 2016, "total": 7448332, "homes": 3652339, "dones": 3795993,
  "year": 2017, "total": 7496276, "homes": 3673224, "dones": 3823052,
  "year": 2018, "total": 7543825, "homes": 3694002, "dones": 3849823,
  "year": 2019, "total": 7619404, "homes": 3732732, "dones": 3886672,
  "year": 2020, "total": 7722203, "homes": 3787684, "dones": 3934519,
  "year": 2021, "total": 7739758, "homes": 3797115, "dones": 3942643,
  "year": 2022, "total": 7747709, "homes": 3799485, "dones": 3948224
}
```

(l'alumne/a adjunta la captura de pantalla)

## Pregunta 3 (2,5 punts). Acabant el controlador del servei RESTful

Encara que les dades amb què treballem són bàsicament estàtiques (no varien amb el temps), volem implementar, ni que sigui per motius didàctics, les operacions CRUD.

La lectura ja la tenim feta amb els endpoints */years* i */years/{year}*. A *YearRepository* afegim els mètodes:

```
void add(Year year);
void update(Year year); (modifica qualsevol dels valors de població, però no l'any)
void delete(int year); (no li passem l'objecte, sinó el valor que representa l'any)
```


i els implementem a *InMemoryYearRepository.java*

Compte no confondre's perquè, segons el context, la variable *year* pot representar l'objecte *Year*, o bé pot representar l'any (el valor numèric). Pots utilitzar els teus noms de variables, però no és factible utilitzar *any* (paraula reservada) ni *año* (caràcter no vàlid). Sí que podríem utilitzar *ano* :-)

Tenim els endpoints de lectura, i els podem testejar amb *curl* (o postman):

```
$ curl http://localhost:8080/poblaciocatalunya/years

[{"year":1981,"total":5949829,"homes":2916527,"dones":3033302},
```

	Codi: I71	Exercici d'avaluació continuada 6	Pàgina de
	Versió: 01	DAW_M07B2_EAC6_Enunciat_2223S2	Lliurament: 16/05/2023

```
{ "year": 1982, "total": 5962723, "homes": 2923258, "dones": 3039465 },  
{ "year": 1983, "total": 5974821, "homes": 2928748, "dones": 3046073 },  
{ "year": 1984, "total": 5978426, "homes": 2929616, "dones": 3048810 },  
...  
]  
  
$ curl http://localhost:8080/poblaciocatalunya/years/2005  
  
{ "year": 2000, "total": 6174547, "homes": 3015079, "dones": 3159468 }
```

### (l'alumne/a adjunta la captura de pantalla)

Fixa't bé que *years/* ens retorna un array d'objectes, mentre que *years/{year}* ens retorna directament l'objecte.

Recorda que les operacions de crear, modificar i esborrar les haurem fet amb el RequestMethod de *POST*, *PUT* i *DELETE* respectivament. Podem testejar el funcionament correcte d'aquestes operacions amb *curl*:

NOTA. Si segueixes els apunts del mòdul (exemple de *Team* i *Player*), el nostre problema és més senzill, però hi ha un problema en convertir la cadena JSON que representa l'objecte *Year* a l'objecte *Year*, segurament perquè les dades són números enters en comptes de tot cadenes. És millor passar el String en format JSON, i a dins del *create* parsejar la cadena i extreure els camps *year*, *total*, *homes* i *dones*.

```
//ResponseEntity<Year> create(@RequestBody Year year) { // no funciona,  
// problema en fer el POST i PUT  
// ResponseEntity<Year> create(@RequestBody String stryear) {
```

Aleshores haurem de convertir el *String* (que no donarà problemes) a objecte *JSON*. Ho pots fer per exemple mirant:

- <https://www.javatpoint.com/how-to-convert-string-to-json-object-in-java>

Hem d'afegir la dependència

```
<dependency>  
  <groupId>org.json</groupId>  
  <artifactId>json</artifactId>  
  <version>20230227</version>  
</dependency>
```

i aleshores ja tindrem disponible un objecte *Year*:

```
JSONArray array = new JSONArray(stryear);  
JSONObject object = array.getJSONObject(0);
```

i per accedir al valor de l'any fem:

```
Integer.parseInt(object.getString("year"))
```

Si ho fem d'aquesta manera hem de passar com a *body* de la petició un array d'un sol element. Finalment, les peticions de creació i modificació serien:

```
$ curl -H "Content-Type: application/json" -X POST -d ' [{"year": "2024",  
"total": "5000000", "homes": "2000000", "dones": "3000000"} ] '  
http://localhost:8080/poblaciocatalunya/years  
  
$ curl -H "Content-Type: application/json" -X PUT -d ' [{"year": "2000",  
"total": "5000000", "homes": "2000000", "dones": "3000000"} ] '  
http://localhost:8080/poblaciocatalunya/years  
  
$ curl -X DELETE http://localhost:8080/poblaciocatalunya/years/2005
```

(l'alumne/a adjunta la captura de pantalla)

### Pregunta 4 (1 punt). Creació d'un client Java *stand-alone* mínim

Ara que ja tenim els serveis web RESTful, volem consumir aquests serveis. En aquest apartat farem un client JAVA, i en el següent apartat farem un client HTML/Javascript.

Afegim la dependència:


```
<dependency>  
<groupId>com.fasterxml.jackson.core</groupId>  
<artifactId>jackson-databind</artifactId>  
<version>2.5.3</version>  
</dependency>
```

Creem el paquet *cat.xtec.ioc.client*. Creem la classe *YearClient*, que modelarà la resposta del servei web, i la classe *Main*, que serà pròpiament el client.

L'objectiu, amb finalitats purament didàctiques, és visualitzar la població que tenia Catalunya l'any 2000. Fem el **Run File** sobre el fitxer *Main.java*, i hem de veure per la consola la població l'any 2000.

```
-----< cat.xtec.ioc.poblaciocatalunya >-----  
] Building poblaciocatalunya 1.0-SNAPSHOT  
-----[ war ]-----  
  
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ poblaciocatalunya ---  
· Informació {year='2000', total=6174547}  
-----  
BUILD SUCCESS  
-----  
Total time: 0.889 s  
Finished at: 2023-03-22T09:51:03+01:00  
-----
```

(l'alumne/a adjunta la captura de pantalla)

	Codi: I71	Exercici d'avaluació continuada 6	Pàgina de
	Versió: 01	DAW_M07B2_EAC6_Enunciat_2223S2	Lliurament: 16/05/2023

## Pregunta 5 (2,5 punts). Client HTML/Javascript. Highcharts.com

Per visualitzar les dades en forma de gràfica i interactiva farem servir la llibreria **Highcharts**:

- <https://www.highcharts.com>

En l'apartat Demo de la seva pàgina web podem veure un portfoli ample del tipus de gràfiques que podem realitzar amb aquesta llibreria.

Des de Javascript, ens interessa accedir a les dades de forma asíncrona (AJAX), fent una petició al servidor web on tenim implementada la API. Un cop recuperem les dades en format JSON podrem construir la nostra gràfica. Volem utilitzar la petició AJAX de la forma més nativa possible (objecte *XMLHttpRequest*), sense utilitzar altres llibreries ni frameworks que afegirien soroll. Se't proporciona la part del client ja finalitzada (aquesta funcionalitat correspon al mòdul M06 corresponent a Programació Web en el cantó de client):

- script *poblacio\_catalunya.html*

Per fer funcionar el codi html, podríem integrar el fitxer html dins del projecte JAVA (per exemple en la carpeta WEB-INF/). Ara bé, preferim separar la part de servidor i la part de client, simulant que són dues funcionalitats separades. En principi necessitaríem un servidor web com Apache funcionant pel port 80:

- [http://localhost/poblaciocatalunya/poblacio\\_catalunya.html](http://localhost/poblaciocatalunya/poblacio_catalunya.html)

Ara bé també funcionarà obrint directament el script en el navegador:

- [file://ruta/poblaciocatalunya/poblacio\\_catalunya.html](file://ruta/poblaciocatalunya/poblacio_catalunya.html)

Tant en un cas com en l'altre tenim un problema: *"request has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource."*

No és un problema, és un tema de seguretat. Els servidors web d'entrada no permeten peticions que provinquin d'un domini diferent. El nostre servei web el tenim funcionant a localhost, però pel port 8080, i una petició que provinguí de localhost:80 es considera que és un domini diferent.

Per tant, hem d'habilitar en el nostre servidor *Glassfish* la possibilitat de tramitar peticions que vinguin de dominis remots. Són canvis que s'han de fer en la part del servidor, no en la part del client.


Una manera de fer-ho (<https://coderanch.com/t/640189/java/Access-Control-Origin-header-present>) és:

1) afegir una dependència al pom.xml:

```
<dependency>
  <groupId>com.thetransactioncompany</groupId>
  <artifactId>cors-filter</artifactId>
  <version>1.7</version>
</dependency>
```

2) Afegir un *filter* i un *filter-mapping* a *web.xml*:

```
<filter>
  <filter-name>CORS</filter-name>
  <filter-class>com.thetransactioncompany.cors.CORSFilter</filter-class>
```

	Codi: I71	Exercici d'avaluació continuada 6	Pàgina de
	Versió: 01	DAW_M07B2_EAC6_Enunciat_2223S2	Lliurament: 16/05/2023



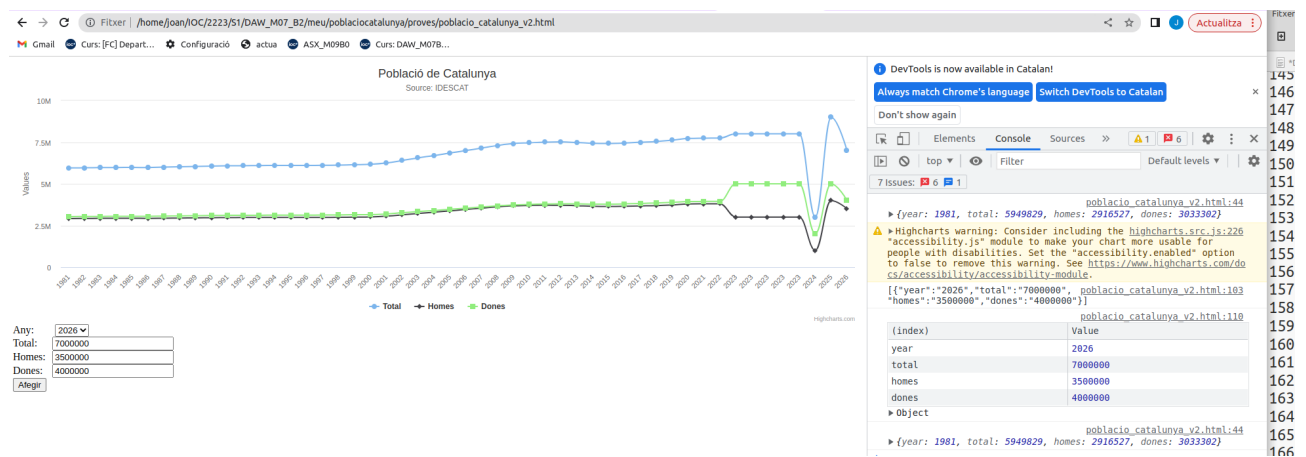
## Formació professional

```
<init-param>
  <param-name>cors.allowOrigin</param-name>
  <param-value>*</param-value>
</init-param>
<init-param>
  <param-name>cors.supportedMethods</param-name>
  <param-value>GET, POST, HEAD, PUT, DELETE</param-value>
</init-param>
<init-param>
  <param-name>cors.supportedHeaders</param-name>
  <param-value>Accept, Origin, X-Requested-With, Content-Type,
    Last-Modified</param-value>
</init-param>
<init-param>
  <param-name>cors.exposedHeaders</param-name>
  <param-value>Set-Cookie</param-value>
</init-param>
<init-param>
  <param-name>cors.supportsCredentials</param-name>
  <param-value>true</param-value>
</init-param>
</filter>

<filter-mapping>
  <filter-name>CORS</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Després de fer aquests canvis (reiniciar el servidor web i fer *clean & build*) ha de funcionar.


Mireu amb deteniment el script *poblacio\_catalunya.html*. No només s'ha implementat la petició AJAX i la construcció de la gràfica, sinó que també s'ha volgut implementar un formulari per afegir nous anys (2023-2026), amb la petició *POST* i les dades que recollim del formulari.



(l'alumne/a adjunta la captura de pantalla)

**Tasca voluntària:** un cop funciona, implementa la **modificació** de les dades d'un any, i l'**eliminació** d'un any.

(l'alumne/a adjunta la captura de pantalla)

	Codi: I71	Exercici d'avaluació continuada 6	Pàgina de
	Versió: 01	DAW_M07B2_EAC6_Enunciat_2223S2	Lliurament: 16/05/2023