

# Ramos Da Fonte, Alex

## PEC 1

### M4.257 - Eines HTML i CSS

#### Aula 1

Github: <https://github.com/AlexFonte/EINES-I-HTML-CSS-PAC-1>

Publicació Web: <https://aramosda-eineis-1-pec.netlify.app/>

## Index

1. Creació del boilerplate basat en Parcel. Com s'ha definit l'entorn de desenvolupament i producció, i com es configura el suport a navegadors antics. ....	3
2. Gestió de dependències: pre- o postprocessadors, i dependències addicionals. ....	5
3. Creació del repositori Git .....	10
4. Adequació a la temàtica i estructura de la pràctica.....	11
5. Disseny responsive, complexitat i estètica .....	12
6. Publicació a internet .....	14
7. Enllacos .....	14

## 1. Creació del boilerplate basat en Parcel. Com s'ha definit l'entorn de desenvolupament i producció, i com es configura el suport a navegadors antics.

El pas consisteix a crear una base de projecte, com si fos una plantilla. En aquest cas, utilitzarem **npm** en lloc de **yarn**, ja que és l'eina que conec millor.

Per començar, instal·lem Parcel com a dependència de desenvolupament amb la següent comanda:

```
npm install --save-dev parcel
```

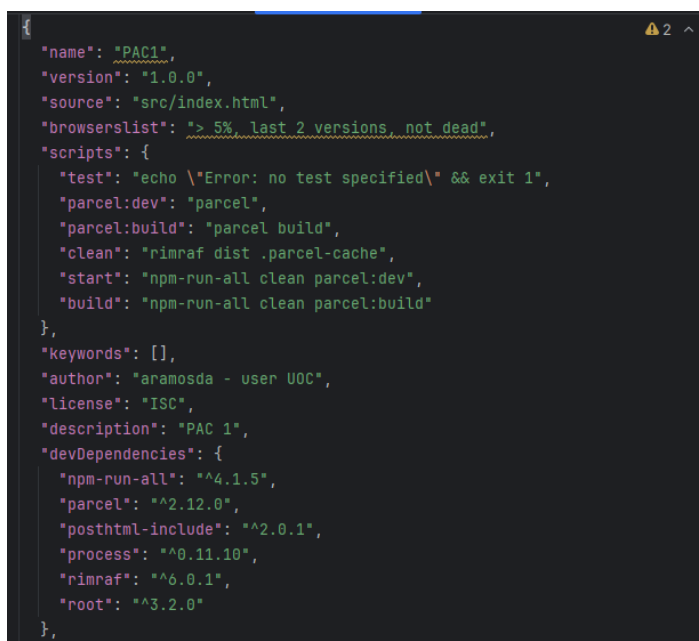
Això ens permet crear una plantilla base o un **boilerplate** de Parcel.

Un cop creat el projecte, podem començar a crear els nostres fitxers. Parcel ja proporciona les comandes necessàries per treballar en diferents entorns:

Per a l'entorn de desenvolupament `npx parcel nom_fitxer`

Per a l'entorn de producció: `npx parcel build nom_fitxer`

Tanmateix, utilitzar el fitxer **package.json** ens farà més fàcil gestionar i configurar els entorns de treball.



```
{
  "name": "PAC1",
  "version": "1.0.0",
  "source": "src/index.html",
  "browserslist": "> 5%, last 2 versions, not dead",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "parcel:dev": "parcel",
    "parcel:build": "parcel build",
    "clean": "rimraf dist .parcel-cache",
    "start": "npm-run-all clean parcel:dev",
    "build": "npm-run-all clean parcel:build"
  },
  "keywords": [],
  "author": "aramosda - user UOC",
  "license": "ISC",
  "description": "PAC 1",
  "devDependencies": {
    "npm-run-all": "^4.1.5",
    "parcel": "^2.12.0",
    "posthtml-include": "^2.0.1",
    "process": "^0.11.10",
    "rimraf": "^6.0.1",
    "root": "^3.2.0"
  }
}
```

Aquí podem veure com, als scripts, tenim definit **el start** per a l'entorn de desenvolupament i **el build** per a l'entorn de producció. Aquestes dues comandes ens serviran per mantenir separats els entorns.

Dins de les comandes, tenim un **clean**, que serveix per netejar la memòria cau de Parcel, ja que, de tant en tant, tenia el problema que no es recarregava el fitxer. Amb la comanda **clean**, cada vegada que executem `npm run start` o `npm run build`, s'eliminarà la carpeta **dist**, on Parcel genera els seus fitxers minimitzats i comprimits.

Per poder utilitzar la comanda **clean**, he hagut de fer:

```
npm install --save-dev rimraf npm-run-all
```

i afegir les següents línies a **scripts** dins del package.json:

```
"parcel:dev": "parcel",
"parcel:build": "parcel build",
"clean": "rimraf dist .parcel-cache",
"start": "npm-run-all clean parcel:dev",
"build": "npm-run-all clean parcel:build"
```

Amb això, ja queda resolt el problema de la memòria cau de Parcel.

D'altra banda, Parcel té suport per a diferents versions de navegadors antics, afegint la línia "browserslist": "" al package.json. Dins d'aquesta línia, podem definir, amb paràmetres, fins on volem la compatibilitat de la nostra pàgina. En el meu cas, he configurat compatibilitat amb navegadors que siguin utilitzats per més del 5% dels usuaris, o amb les últimes 5 versions de cada navegador, i que no estiguin obsolets o descontinuats.

## 2. Gestió de dependències: pre- o postprocessadors, i dependències addicionals.

Per a les dependències addicionals, he decidit començar amb **FontAwesome**, ja que disposa d'una versió gratuïta amb icones molt útils. D'aquesta manera, no caldrà buscar icones gratuïtes per internet.

Per utilitzar aquesta dependència, seguirem aquests passos:

- Instal·lar-la: `npm i --save @fortawesome/fontawesome-free`
- Importar el CSS de FontAwesome al nostre fitxer CSS: `@import "../node_modules/@fortawesome/fontawesome-free/css/all.min.css";`
- Finalment, ja podrem utilitzar les icones en el nostre projecte amb etiquetes com aquesta: `<i class="fa-brands fa-facebook fa-xl"></i>`

D'altra banda, també he decidit instal·lar la dependència **Hover.css**, que proporciona efectes interactius sobre elements HTML com botons, imatges, div, etc. He optat per aquesta dependència perquè ofereix l'efecte **zoom** (anomenat "grow") que necessitava, encara que també integra altres efectes interessants.

Tot i que es podria fer manualment amb CSS, aprofitar aquesta dependència estalvia temps. Els passos per utilitzar-la són:

- Instal·lar-la: `npm i --save hover.css`
- Importar el CSS de Hover al nostre fitxer CSS: `@import "../node_modules/hover.css/css/hover.css";`
- Utilitzar l'efecte desitjat, per exemple, afegint la classe **hvr-grow** a l'etiqueta HTML corresponent: `<button class="hvr-grow">Clica'm</button>`

Finalment, he instal·lat **SweetAlert2**, una dependència que permet crear finestres emergents (pop-ups) amb dissenys personalitzables per a missatges d'èxit, error, informació, avís, etc. Encara que la predecessora **SweetAlert** també és funcional, SweetAlert2 ofereix més opcions de configuració. Per utilitzar-la, cal seguir aquests passos:

- Instal·lar-la: `npm i --save sweetalert2`
- Importar la dependència en un fitxer .js: `import Swal from 'sweetalert2'`. **Nota:** Això només funcionarà si el fitxer .js s'importa al HTML com a mòdul: `<script type="module" src="js/app.js"></script>`
- Un cop feta la importació, podem crear el pop-up segons les necessitats del projecte.

```
const sweet : {...} = Swal.mixin( options: {
  customClass: {
    container: "background-viana-hello",
    popup: "background-viana-hello-popup",
    confirmButton: "background-viana-hello-btn",
  }
});

sweet.fire( options: {title:'Benvingut Viana do Castelo: Una Joia Històrica i Cultural de Portugal'});
```

Al final haurem de tenir el resultat següent en el package.json

```
"dependencies": {  
  "@fortawesome/fontawesome-free": "^6.6.0",  
  "hover.css": "^2.3.2",  
  "sweetalert2": "^11.14.5"  
}
```

Ara parlarem dels **preprocessadors** i **postprocessadors**.

En primer lloc, cal destacar que **Parcel** ja inclou **PostCSS** per defecte. Això significa que, quan hem fet l'**import** de **Hover.css** i **FontAwesome** dins del nostre fitxer CSS, en executar `npm run start` o `npm run build`, Parcel processarà i **autoprefixarà** automàticament tot el contingut.

Aquest procés inclou incorporar els imports directament dins del fitxer CSS generat per Parcel, que es troba a la carpeta **dist/\*.css**.

El resultat final passarà de tenir aquest format (abans d'utilitzar PostCSS):

css

```
.example {  
  display: flex;  
}
```

A tenir aquest format, després de ser processat:

css

```
.example {  
  display: -webkit-box;  
  display: -ms-flexbox;  
  display: flex;  
}
```

D'aquesta manera, **PostCSS** assegura que el CSS generat sigui compatible amb la majoria de navegadors, afegint prefixos automàticament quan sigui necessari.

```

1  @import "../../node_modules/@fortawesome/fontawesome-free/css/all.min.css";
2
3  @import "../../node_modules/hover.css/css/hover.css";
4
5
6  :root {
7    --v-red: #E01846;
8    --v-gold: #D6B76F;
9    --onest-sans-serif: 'Onest';
10   --custom-gray: #f2f2f2;
11   --custom-white: #fefefe;
12   --box-sadow: 3px 3px 5px rgba(214, 183, 111, 0.85);
13 }
14
15 .arizonia {
16   font-family: "Arizonia", cursive;
17   font-weight: 400;
18   font-style: normal;
19 }
20
21 .gold {
22   color: var(--v-gold);
23 }
24

```

Després de executar `npm run build`, respectant l'ordre dels imports així com el nostre CSS.

```

/*
 * Hover.css (http://ianlunn.github.io/Hover/)
 * Version: 2.3.2
 * Author: Ian Lunn @IanLunn
 * Author URL: http://ianlunn.co.uk/
 * Github: https://github.com/IanLunn/Hover
 *
 * Hover.css Copyright Ian Lunn 2017. Generated with Sass.
 */
/*
 * Font Awesome Free 6.0.0 by @fontawesome - https://fontawesome.com
 * License - https://fontawesome.com/license/free (Icons: CC BY 4.0, Fonts: SIL OFL 1.1, Code: MIT License)
 * Copyright 2024 Fonticons, Inc.
 */
.hvr-grow{vertical-align:middle;transition-property:transform;transition-duration:.3s;display:inline-block;transform:perspective(1px)translateZ(0);box-shadow:0 0 1px #0000}.hvr-grow:active{transform:scale(1.1)}.hvr-shrink{vertical-align:middle;transition-property:transform;transition-duration:.3s;display:inline-block;transform:perspective(1px)translateZ(0);box-shadow:0 0 1px #0000}.hvr-shrink:active{transform:scale(.9)}@keyframes hvr-pulse{25%{transform:scale(1.1)}75%{transform:scale(.9)}}.hvr-pulse{vertical-align:middle;display:inline-block;transform:perspective(1px)translateZ(0);box-shadow:0 0 1px #0000}.hvr-pulse:active{transform:scale(1.1)}.hvr-pulse-grow{vertical-align:middle;display:inline-block;transform:perspective(1px)translateZ(0);box-shadow:0 0 1px #0000}.hvr-pulse-grow:active{transform:scale(1.1)}.hvr-pulse-grow:active{animation-name:hvr-pulse-grow;animation-duration:.3s;

```

Bàsicament, el que fa és incorporar el codi dels CSS de tercers dins del nostre propi CSS. Hem de tenir una cosa clara: **PostCSS**, que s'inclou amb Parcel, es veu afectat per l'ús i l'especificitat definida a **browserslist**.

D'altra banda, també he utilitzat un **postprocessador** anomenat **PostHTML**, que Parcel incorpora per defecte. Aquest permet afegir configuracions específiques per incloure un fitxer HTML dins d'un altre, funcionant com un mòdul HTML.

Per configurar-ho, cal fer el següent:

- Instal·lar el paquet necessari: `npm i --save posthtml-include`
- Crear el fitxer **.posthtmlrc** a l'arrel del projecte.
- Afegir-hi la següent configuració, que resol el problema de localitzar els fitxers HTML a incorporar si no es troben al mateix directori:

```

{
  "plugins": {

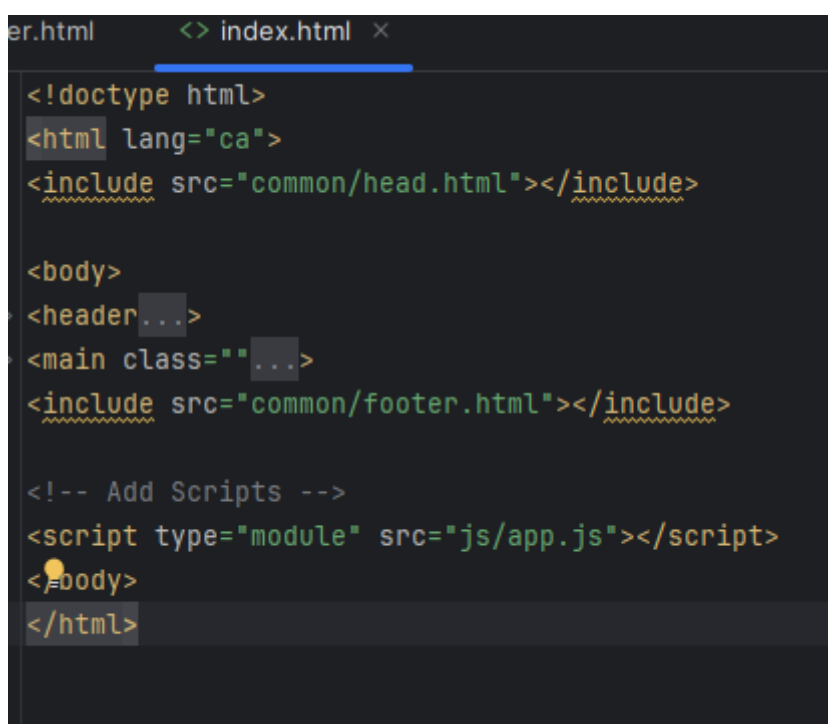
```

```
"posthtml-include": {
  "root": "src/"}}
```

Un cop fet això, per incorporar HTML dins d'un altre fitxer, seguirem aquests passos:

- Crear un fitxer **footer.html** dins de **src/common** (aquesta ruta s'utilitzarà per incloure tot el contingut HTML que sigui repetitiu, com ara el codi del footer).
- Al fitxer **index.html**, afegir aquest HTML mitjançant l'etiqueta següent:  
`<include src="common/footer.html"></include>`

Per defecte, a la versió de desenvolupament, el resultat que veurem serà el següent:



```
er.html  <> index.html x
<!doctype html>
<html lang="ca">
<include src="common/head.html"></include>

<body>
<header...>
<main class="...">
<include src="common/footer.html"></include>

<!-- Add Scripts -->
<script type="module" src="js/app.js"></script>
</body>
</html>
```

Però, una vegada postprocessat amb **PostHTML**, a la versió de producció, si ens fixem en el fitxer **index.\*.html** que es troba a la ruta **dist/**, observarem el següent resultat: el codi del **footer** s'ha incorporat directament dins del fitxer **index.html**.

Aquest comportament és molt útil, ja que evita la necessitat de carregar fitxers addicionals i assegura que tot el contingut repetitiu (com el footer) quedi integrat al document final, optimitzant el rendiment de la pàgina en producció.



&lt;&gt; dist\index.html x

```

1  <!doctype html>
2  <html lang="ca">
3  > <head>
15 <body>
16 > <header>
34 > <main>
68 <footer>
69     <div class="container">
70         <div class="footer">
71             <div class="footer-logo-autor">
72                 <div><a class="logo-img" href="/index.html"></a></div>
73                 <div class="author"><p>2024 © Alex Ramos Da Fonte</p></div>

```

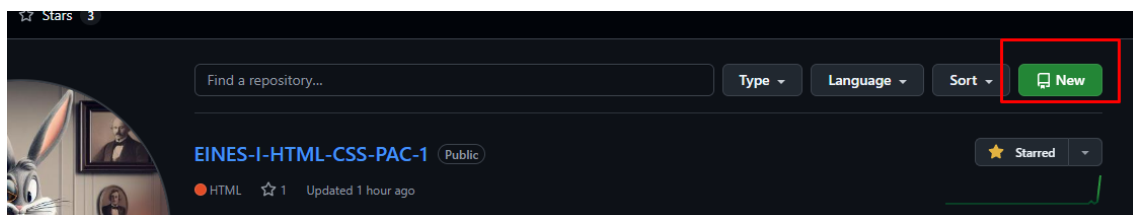
### 3. Creació del repositori Git

Per a la creació del repositori **Git**, he utilitzat el meu compte privat de GitHub:  
<https://github.com/AlexFonte>.

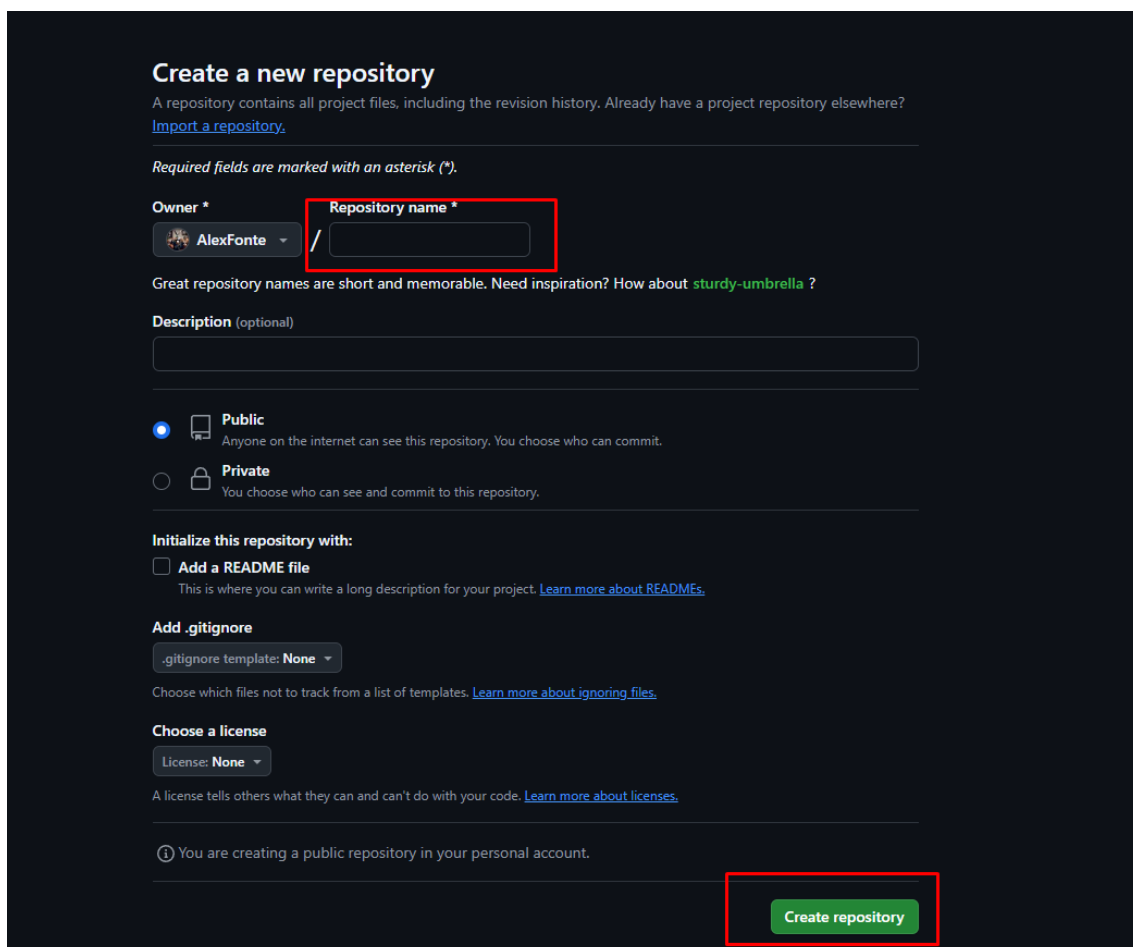
**GitHub** és una eina gratuïta per al control de versions de codi, tot i que també existeixen altres alternatives com **GitLab**.

Atès que **GitHub** permet crear repositoris directament des de la seva interfície gràfica, he seguit els passos següents:

Accedir a la meua compte de GitHub.



Afegir un nom de repositori i create repository:



D'aquesta manera, tindrem el repositori creat i ja disposarem de la URL per fer commits:

<https://github.com/AlexFonte/EINES-I-HTML-CSS-PAC-1.git>.

Aquesta URL ens permet incorporar el repositori amb un gestor de control de versions com **Sourcetree** o utilitzar les comandes de Git al terminal (cmd). Personalment, prefereixo utilitzar **Sourcetree** o l'eina de Git que incorpora **WebStorm** de JetBrains, ja que ofereixen una interfície gràfica molt intuïtiva per gestionar el codi i les versions.

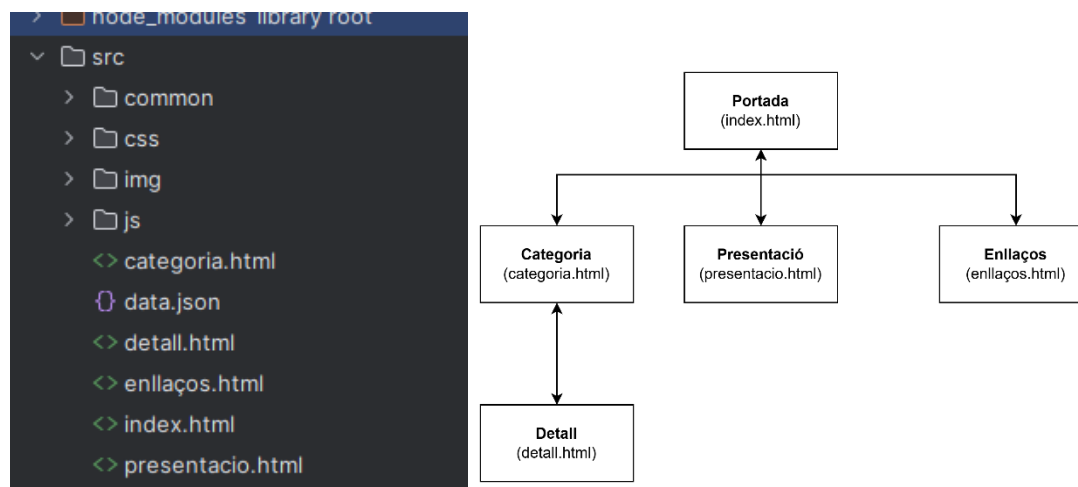
## 4. Adequació a la temàtica i estructura de la pràctica

Per a la temàtica de la pràctica, he decidit escollir una ciutat del nord de Portugal: **Viana do Castelo**, on vaig cada estiu.

**Viana do Castelo** té molt a oferir, tant pel que fa a llocs per visitar com per gaudir o passar-ho bé. Encara que la ciutat sembli petita, als voltants, és a dir, dins del districte de Viana do Castelo, també podem trobar autèntiques joies: monuments, paisatges únics o l'oportunitat d'aprendre alguna curiositat específica d'un petit poble amagat.

A l'estiu, especialment, es pot veure molta gent fent turisme a **Viana do Castelo**. Encara que la major part sigui turisme nacional, moltes persones del sud de Portugal viatgen al nord per conèixer aquesta ciutat. A més, **molts habitants de la zona de Galícia** també mostren interès per visitar-la, reforçant el seu atractiu com a destinació propera.

L'estructura de la pràctica és la següent i serà la mateixa per als fitxers **HTML**.



- **Portada (index.html):**

Aquí trobarem una petita presentació que ens convidarà a descobrir **Viana do Castelo** amb un eslògan de la ciutat que diu: *"Quem gosta vem, Quem ama fica."* (que vol dir: *"Qui li agrada ve, qui l'estima es queda"*).

A més de l'eslògan, hi haurà accessos directes cap a les seccions **Categoria**, **Presentació** i **Enllaços**, sense oblidar el menú principal.

- **Categoria (categoria.html):**  
En aquesta pàgina, es mostrarà un llistat amb punts d'interès, llocs per visitar i plats típics de la ciutat o la regió.
- **Detall (detall.html):**  
Aquí es podrà veure el detall del punt d'interès seleccionat. Es descriurà la seva ubicació, què es pot fer allà i una breu història relacionada.
- **Presentació (presentacio.html):**  
En aquesta secció, es trobarà la història i el context de la pàgina web, així com una explicació de com navegar-hi.
- **Enllaços (enllaços.html):**  
En aquesta pàgina, es recolliran tots els enllaços als recursos multimèdia i d'informació utilitzats, incloent-hi el corresponent **copyright**.

#### Organització del codi:

- Totes les imatges es trobaran a **src/img**.
- Les pàgines HTML es trobaran a **src/**.
- Els fitxers CSS es trobaran a **src/css**.
- Els fitxers JavaScript es trobaran a **src/js**.
- Els components compartits, com el **footer** i el **head**, es trobaran a **src/common/**

## 5. Disseny responsive, complexitat i estètica

### Estètica de la pàgina

He decidit utilitzar **3 o 4 colors principals**:

- **Daurat (--v-gold: #D6B76F):** Aquest color representa **Viana do Castelo** amb la seva filigrana d'or i el cor d'or. Es troba en elements com el menú, el footer, els botons i els punts de les llistes (*dots*).
- **Vermell (--v-red: #ED1846):** També simbolitza la ciutat per les seves festes i vestits tradicionals. S'utilitza en els estats *hover* sobre imatges, botons i al menú (en l'element actiu).
- **Escala de grisos:** S'utilitza per al footer, el menú i per separar seccions horitzontals.
- **Blanc:** El color del fons de la pàgina (body).

## Tipografia

He triat **dos tipus de fonts** per la pàgina:

- **Arizonia** per als títols: Una lletra lligada que recorda la filigrana de Viana, com el filferro.
- **Onest** per al text general: És clara i fàcil de llegir.

## Estructura visual

- Quan hi ha diverses seccions, se separen amb colors alternants entre blanc i gris.
- Si hi ha text a l'esquerra i una imatge a la dreta en una secció, la següent secció tindrà el text a la dreta i la imatge a l'esquerra.
- En versions mòbils o de *tablet*, totes les seccions es disposen en format vertical amb text seguit d'imatge, per facilitar la lectura.

## Imatge destacada

A cada pàgina, sota el menú, hi haurà una **imatge d'amplada completa (100%)** amb un títol escrit amb la font **Arizonia**, ocupant un 80% de l'amplada, en color blanc per millorar la llegibilitat.



## Responsive Design

Per aconseguir un disseny adaptable (*responsive*), s'ha treballat amb **flexbox** o **CSS Grid**, que faciliten l'aplicació d'estils segons els punts de ruptura (*breakpoints*):

- **Grilles dinàmiques:**
  - 1 columna en versions mòbils.
  - 2 columnes per a *tablets*.
  - 4 columnes per a pantalles d'ordinador (petites o grans).
- **Breakpoints definits:**
  - @media (max-width: 600px) → Versió mòbil.
  - @media (min-width: 601px) and (max-width: 991px) → Versió tablet.
  - @media (min-width: 992px) → Ordinadors amb pantalles petites.
  - @media (min-width: 1200px) → Pantalles grans.

Per a pantalles amb un mínim de **992px d'amplada**, el contingut mai ocuparà el 100% del *width*. El màxim serà un **80% de l'amplada** per mantenir una estètica centrada i

equilibrada. En canvi, per pantalles petites, intentarem ocupar tot l'espai disponible, evitant l'*scroll* lateral.

### Desenvolupament del disseny responsive errades.

La part més complicada ha estat fer que el disseny sigui **totalment responsive**, ja que vaig començar amb un enfocament *desktop-first* en lloc de *mobile-first*, que hauria facilitat la tasca.

## 6. Publicació a internet

He decidit utilitzar **Netlify**, ja que és una eina que vam veure al curs. Tot i que també conec altres opcions com **GitHub Pages** o **Vercel**, que permeten la publicació a internet i generen un URL per fer accessible el projecte al públic, vaig voler provar **Netlify** per veure com funciona.

Tant netlify com github pages els 2 s'autoconfigurem per segons el projecte per publicar a la web.

Per començar, vaig fer login amb el meu compte de **GitHub**. Un cop fet el login, vaig veure que la facilitat que ofereix **Netlify** per publicar un projecte simplement seleccionant el repositori de GitHub va ser molt còmoda i senzilla.

Tot i això, vaig tenir un petit problema amb la versió de **Node.js** que utilitza Netlify per defecte. Per defecte, Netlify utilitza la versió **14.x**, però jo havia desenvolupat el projecte amb la versió LTS **20.18.x**. Quan vaig intentar compilar, vaig obtenir un error. La solució va ser senzilla: vaig modificar la configuració de **Node.js** a **Netlify**, indicant que volia utilitzar la versió **20.18.x**. A més, vaig afegir la línia següent al fitxer **package.json** per forçar l'ús d'aquesta versió de **Node**:

```
"engines": {  
  "node": "20.18.0"  
}
```

Afegint aquest canvis el projecte a netlify, compila sense cap problema.

## 7. Enllacos

- **Imatges:**
  - **Autor:** olhar viana do castelo,  
**Enllaç:** <https://www.olharvianadocastelo.pt/2016/03/sao-cerca-de-quarenta-os-restaurantes.html>
  - **Autor:** VIANA COOL,  
**Enllaç:** <https://vianacool.com/en/eat/confeitaria-manuel-natario>
  - **Autor:** VIANA FESTAS,  
**Enllaç:** <https://festasdagonia.com/a-romaria/desfile-da-mordomia/>

- **Autor:** Osvaldo Gago,  
**Enllaç:** [https://commons.wikimedia.org/wiki/File:Estacao\\_Comboio\\_Viana\\_Castelo.JPG](https://commons.wikimedia.org/wiki/File:Estacao_Comboio_Viana_Castelo.JPG)
- **Autor:** VIANA FESTAS,  
**Enllaç:** <https://festasdagonia.com/en/a-romaria/desfile-da-mordomia/>
- **Autor:** Xosema,  
**Enllaç:** [https://commons.wikimedia.org/wiki/File:Viana\\_do\\_Castelo\\_-\\_Buque\\_Gil\\_Eannes\\_-\\_01.jpg](https://commons.wikimedia.org/wiki/File:Viana_do_Castelo_-_Buque_Gil_Eannes_-_01.jpg)
- **Autor:** png design,  
**Enllaç:** [https://unsplash.com/es/fotos/persona-usando-la-camara-plegable-CSWcbb\\_FriU](https://unsplash.com/es/fotos/persona-usando-la-camara-plegable-CSWcbb_FriU)
- **Autor:** Viver Viana,  
**Enllaç:** <https://viverviana.pt/viana-do-castelo/>
- **Autor:** A Casa do Mestre,  
**Enllaç:** <https://acasadomestre.com/pastelarias-e-doces-de-viana/>
- **Autor:** Camara Municipal Viana do Castelo,  
**Enllaç:** <https://www.cm-viana-castelo.pt/visitar/turismo-e-lazer/museus-e-espacos-museologicos/museu-do-traje/apresentacao/>
- **Autor:** Roberto Santorini,  
**Enllaç:** [https://commons.wikimedia.org/wiki/File:Ponte\\_Eiffel\\_em\\_Viana\\_do\\_Castelo.jpg](https://commons.wikimedia.org/wiki/File:Ponte_Eiffel_em_Viana_do_Castelo.jpg)
- **Autor:** Joseolgon,  
**Enllaç:** [https://commons.wikimedia.org/wiki/File:Aerial\\_Ponte\\_de\\_Lima.jpg](https://commons.wikimedia.org/wiki/File:Aerial_Ponte_de_Lima.jpg)
- **Autor:** Camara Municipal Viana do Castelo,  
**Enllaç:** <https://www.cm-viana-castelo.pt/locais/praias-do-cabedelo/>
- **Autor:** olhar viana do castelo,  
**Enllaç:** <https://www.olharvianadocastelo.pt/2019/01/viana-santa-luzia-vista-de-outra.html>
- **Autor:** Afonso Matos,  
**Enllaç:** <https://unsplash.com/es/fotos/un-gran-edificio-de-piedra-CffeVaem-U0>
- **Autor:** olhar viana do castelo,  
**Enllaç:** <https://www.olharvianadocastelo.pt/2019/01/viana-santa-luzia-vista-de-outra.html>
- **Autor:** olhar viana do castelo,  
**Enllaç:** <https://www.olharvianadocastelo.pt/2019/01/viana-santa-luzia-vista-de-outra.html>
- **Autor:** Receitas e Menus,  
**Enllaç:** <https://www.receitasemenus.net/arroz-de-sarrabulho-a-moda-de-ponte-de-lima/>
- **Autor:** Portugal Memories Tour, **Enllaç:** <https://portugalmemoriestour.com/sabia-torta-de-viana-deliciosa-agora-certificada/>

- **Autor:** olhar viana do castelo,  
**Enllaç:** <https://www.olharvianadocastelo.pt/2020/01/viana-do-castelo-e-cidade-ha-172-anos.html>
- **Autor:** Camara Municipal Viana do Castelo,  
**Enllaç:** <https://www.cm-viana-castelo.pt/>
- **Autor:** Bextrel,  
**Enllaç:** [https://commons.wikimedia.org/wiki/File:Vista\\_aerea\\_-\\_Viana\\_do\\_Castelo\\_-\\_29.01.2023.jpg](https://commons.wikimedia.org/wiki/File:Vista_aerea_-_Viana_do_Castelo_-_29.01.2023.jpg)
- **Autor:** RH&XL,  
**Enllaç:** <https://www.flickr.com/photos/hong-xiao/52618653978/in/photolist-2oaJmC3-2q9HWbk-t7tSy-2oQ3tuK-pC3Nxf-XeWFmq-21Qy6ub-9zxPF2-MqVC1i-2dhH2d8-ffwDQf-2oawFst-2oarPiF-2oaJmjh-ffwDYs-HQmNcD-JdDW8b-GC4ejB-WGQVBm-9aYhFC-9aYhyU-Uit4FE-HXiHqC-2oyV7Fy-2pdzUj7-9aYhLY-BSdHud-sy5Wh-G5MUby-7BSr8Y-24LWYGh-XiBHer-sy5Wg-UnfdFh-X6TbXF-8Fcfuk>
- **Infografia :**
  - **Santa Luzia:**  
<https://www.farodevigo.es/visado/portugal/2023/10/06/turismo-portugal-mirador-santa-lucia-viana-castelo-92902746.html>
  - **Santa Luzia:** <https://www.olharvianadocastelo.pt/2017/04/viana-miradouro-de-santa-luzia.html?m=1>
  - **Santa Luzia:** <https://visitaronorte.com/santuاريو-de-santa-luzia/>
  - **Santa Luzia:** <https://vivecamino.com/viana-do-castelo/santuاريو-de-santa-luzia-3649/>
  - **Santa Luzia:**  
[https://pt.wikipedia.org/wiki/Santu%C3%A1rio\\_Diocesano\\_do\\_Sagrado\\_Cora%C3%A7%C3%A3o\\_de\\_Jesus\\_\(Viana\\_do\\_Castelo\)#](https://pt.wikipedia.org/wiki/Santu%C3%A1rio_Diocesano_do_Sagrado_Cora%C3%A7%C3%A3o_de_Jesus_(Viana_do_Castelo)#)
  - **Viana do Castelo:** <https://www.cm-viana-castelo.pt/viver/>
  - **Viana do Castelo:** <https://www.dorigemlusa.pt/visitar/viana-do-castelo/>
  - **Viana do Castelo:** <https://xrei.com/es/viana-do-castelo/#:~:text=Viana%20do%20Castelo%20fue%20fundada,la%20Epopeya%20de%20los%20Descubrimientos>
  - **Viana do Castelo:** [https://pt.wikipedia.org/wiki/Viana\\_do\\_Castelo](https://pt.wikipedia.org/wiki/Viana_do_Castelo)
  - **Viana do Castelo:** <https://www.visitportugal.com/es/destinos/porto-e-norte/73741>
  - **Viana do Castelo:** [https://www.infopedia.pt/artigos/\\$viana-do-castelo#](https://www.infopedia.pt/artigos/$viana-do-castelo#)
  - **Viana do Castelo, plats típics:** <https://www.cm-viana-castelo.pt/visitar/turismo-e-lazer/onde-comer-e-beber/tradicoes-culinarias-locais/>