



**PWA**

# Índex

Introducció.....	1
Què són les PWA.....	1
Com es desenvolupa una PWA.....	1
Creant una PWA.....	1
Requisits.....	1
Creant un projecte.....	1
HTML.....	2
head.....	2
meta.....	2
title.....	2
favicon.....	2
body.....	2
Header.....	2
Slider.....	3
Serveis.....	3
videos.....	4
footer.....	5
CSS.....	6
Estils Generals.....	6
Estils barra navegació i header.....	7
Estils Slider.....	8
Estil Serveis.....	8
Estils Video.....	9
Estils Footer.....	10
Maquetació responsive.....	11
Media Queries.....	11
Maquetació responsive capçalera.....	12
Maquetació responsive slider.....	17
Maquetació responsive serveis i video.....	20
Creant la PWA (Aplicació Web Progressiva).....	23
Etiquetes meta PWA.....	24
Etiqueta color aplicació.....	25
Optimització mobile.....	25
Optimització Apple.....	25
Icona apple.....	25
Manifest.....	25
manifest.json.....	26
Propietat name.....	26
Propietat short_name.....	26
Propietat description.....	26
Propietat background_color.....	26
Propietat theme_color.....	26
Propietat orientation.....	27
Propietat display.....	27
Propietat start_url.....	27
Propietat scope.....	27
Propietat lang.....	27
Propietat icons configuracio icones.....	27
Service Worker.....	29
Creant el Service Worker.....	29

Event install.....	30
Event Activate.....	31
Event fetch.....	32
Publicar PWA servidor web.....	38

# Introducció

## Què són les PWA

Les Progressive Web Application són aplicacions web que comparteixen característiques amb les aplicacions natives, d'aquesta manera una PWA és pot instal·lar en un dispositiu mòbil, simulen el comportament de les aplicacions d'un smartphone normal, es poden executar sense connexió a internet, són ràpides i lleugeres ja que per darrera utilitza un navegador web.

## Com es desenvolupa una PWA

Es desenvolupa com una web normal, es fa una maquetació responsive, es crea un manifest, que és la configuració de la PWA, i s'utilitza un service worker, que és una tecnologia de JavaScript que permet que la PWA funcioni en dispositius mòbils, permetent que la PWA s'executi en segon pla, també ens permet guardar cache al navegador i així no tindre que fer peticions al servidor i fer consultes sense connexió, també permet tindre un events que permetran descarregar la web, un altre guardar la caché i un altre descarregar-se les dades i el contingut per guardar-lo en aquesta caché, també ens permetrà utilitzar l'aplicació sense connexió també es poden enviar notificacions i fer que l'aplicació funcioni.

## Creant una PWA

### Requisits

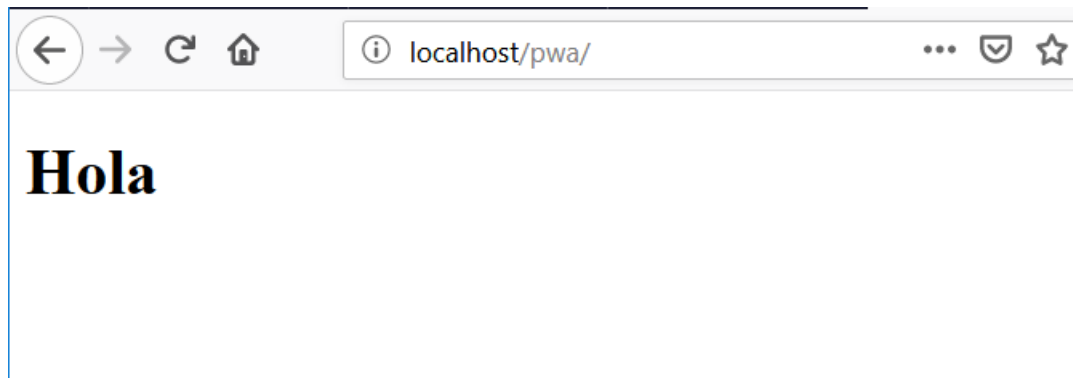
Per crear la nostre PWA necessitarem un navegador web, utilitzarem firefox o chrome, un editor de text, utilitzarem SublimeText i un servidor web local, utilitzarem xamp. També tindrem que tindre instal·lat Git

### Creant un projecte

Crearem una carpeta que contindrà el nostre projecte C:\xampp\htdocs que anomenarem pwa i és on treballarem. Crearem l'arxiu index.html i comprovarem que funciona amb un hola món !



```
index.html x
1
2 <!DOCTYPE html>
3 <html lang="ca">
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>PWA</title>
9 </head>
10 <body>
11   <h1>Hola</h1>
12 </body>
13 </html>
```



## HTML

Crearem el document html de moment, amb les següents etiquetes:

`<html lang="es">` → indica el llenguatge de la pàgina.

### head

dins de l'etiqueta head:

#### meta

`<meta charset="utf-8">` → indicarà la codificació de caràcters de la pàgina: accents ...

#### title

`<title>PWA</title>` → títol de la pàgina en el nostre cas serà PWA.

#### favicon

`<link rel="shortcut icon" type="image/ico" href="img/faviconPWA.ico">` contindrà el favicon de la PWA

### body

Dins de l'etiqueta body hi hauràn les següents etiquetes.

### Header

Utilitzarem la nova etiqueta d'HTML5 header per tindre la nostra capçalera, en el nostre cas dins la capçalera hi haurà un logo i un menú.

El logo i el menú aniràn dins d'un div amb la classe container que després farem servir per centrar el contingut d'aquest.

Tindrem dos div el primer amb `id="logo"` que contindrà una etiqueta `h1` amb el nom de la pàgina.

```
<div id="logo">  
  <h1>Exemple PWA</h1>  
</div>
```

El segon div amb `id="menu"` hi haurà una llista normal i corrent amb el menú

```

<div id="menu">
  <ul>
    <li>
      <a href="#">Inici</a>
    </li>
    <li>
      <a href="#">Serveis</a>
    </li>
    <li>
      <a href="#">Videos</a>
    </li>
    <li>
      <a href="#">contactes</a>
    </li>
  </ul>
</div>

```

## Slider

Crearem una nova secció en un div amb id="slider" que tindrà un div que tindrà la classe container que centrarà el contingut d'aquesta pàgina.

```

<div id="slider">
  <div class="container">
    <h2>Exemple de Progressive Web Apps</h2>
  </div> <!-- tanca container -->
</div> <!-- tanca slider -->

```

## Serveis

Crearem una altra secció en un div amb un id="serveis" amb tres div diferents amb la classe service que contindrà els serveis.

Els div amb la classe service serviràn de contenidor per les següents etiquetes:

<i class="fas fa-grin-tears"></i> → que contindrà una icona.

<h3>Servei grid tears</h3> → que contindrà la capçalera dels serveix.

<p>Text de prova per fer una prova de PWA aquest seria un exemple.</p> → amb un text explicant el servei.

```

<div id="services">
  <h2 class="subheader">Serveis</h2>
  <div class="service">
    <i class="fas fa-grin-beam"></i>
    <h3>Servei grid beam</h3>
    <p>Text de prova per fer una prova de PWA aquest seria un
    exemple.</p>
  </div>

  <div class="service">
    <i class="fas fa-grin-tears"></i>
    <h3>Servei grid tears</h3>
    <p>Text de prova per fer una prova de PWA aquest seria un
    exemple.</p>
  </div>

  <div class="service">
    <i class="fas fa-grin-wink"></i>
    <h3>Servei grid wink</h3>
    <p>Text de prova per fer una prova de PWA aquest seria un
    exemple.</p>
  </div>
</div><!-- tanca service -->

```

## videos

La següent secció serà la de videos, per aquesta secció crearem un div amb id="videos" dins aquest div hi haurà un altre div amb class="container" que ens servirà per centrar el contingut d'aquesta secció.. dins el div amb classe container hi haurà un h2 amb el títol de la secció i tres div amb el següent contingut:

div amb classe video que contindrà  
 iframe → amb un video de youtube  
 parraf → amb la descripció d'aquest video

```

<div id="videos">
  <div class="container">
    <h2 class="subheader">Videos</h2>
    <div class="video">
      <iframe width="560" height="315" src="https://
www.youtube.com/embed/BeH2eH9iPw4" frameborder="0" allow="
accelerometer; autoplay; encrypted-media; gyroscope;
picture-in-picture" allowfullscreen></iframe>
      <p>Videoclip Oficial "Una lluna a l'aigua", cançó que
forma part del tercer disc de Txarango "El Cor de la
Terra", dirigit per Mashup.Barcelona. </p>
    </div>
    <div class="video">
      <iframe width="560" height="315" src="https://
www.youtube.com/embed/9iQwWp3ykRc" frameborder="0" allow="
accelerometer; autoplay; encrypted-media; gyroscope;
picture-in-picture" allowfullscreen></iframe>
      <p>Videoclip Oficial "caminem Lluny", cançó que forma part
del disc de Doctor Prats "Venim de lluny" de la
discogràfica Música Global</p>
    </div>
    <div class="video">
      <iframe width="560" height="315" src="https://
www.youtube.com/embed/Z5LVw2abUlw" frameborder="0" allow="
accelerometer; autoplay; encrypted-media; gyroscope;
picture-in-picture" allowfullscreen></iframe>
      <p>Videoclip Oficial "Fins Que Arribi l'alba", cançó que
forma part del disc dels Catarres "Tots Els Meus
Principis" de la discogràfica Música Global Discogràfica
S.L.</p>
    </div>
  </div> <!-- tanca container -->
</div> <!-- tanca video -->

```

## footer

Crearem una etiqueta footer que contindrà un div amb classe container amb:

h2 → amb el títol de la secció

p → amb un text

una llista amb tres ellaçps amb una icona.

```

<footer>
  <div class="container">
    <h2 class="subheader">Contacte</h2>
    <p>Qualsevol dubte posa't en contacte</p>
    <ul>
      <li>
        <a href="#">
          <i class="fab fa-twitter"></i>
        </a>
      </li>
      <li>
        <a href="#">
          <i class="fab fa-facebook-f"></i>
        </a>
      </li>
      <li>
        <a href="#">
          <i class="fab fa-instagram"></i>
        </a>
      </li>
    </ul>
  </div>
</footer>

```



# CSS

Crearem una carpeta que anomenarem css i dins aquesta hi haurà la nostra fulla d'estils que s'anomenarà styles.css.

L'enllaçarem al nostre document index.html

```
<link rel="shortcut icon" type="image/ico" href="img/favicon.png">
<link rel="stylesheet" type="text/css" href="css/styles.css">
```

## Estils Generals

Crearem uns estils generals que s'aplicaran per tota la pàgina web

\* → amb aquest selector donarem un margin i un padding de 0px; text-decoration none i li direm que el font-family que utilitzarem serà Open Sans i si no ho troba el navegador Verdana

```
/*reset dels marges que posa la web i navegador per defecte*/
*{
  margin: 0px;
  padding: 0px;
  text-decoration: none;
  font-family: "Open Sans", Verdana;
}
```

body → el color de la pàgina serà blanc.

```
/* estils generals per body */
body{
  background: white;
}
```

Crearem dues classes que seran generals per tot el document clearfix i container

```
/*neteja els flotats dels elements */
▼ .clearfix{
  clear: both;
  float: none;
}

/* centrarà els blocks d'aquesta classe */
▼ .container{
  width: 85%; /* ample del block*/
  margin: 0px auto; /* centrar el block*/
}
```

## Estils barra navegació i header

Aquests seran els estils per el logo:.

```
#main-header {  
  background: white;  
  height: 100px;  
  line-height: 85px; /*el text tindrà una altura de línia de 85px*/  
}  
  
#logo {  
  width: 63%; /* es redimensionarà amb la pantalla*/  
  float-left;  
}  
  
#logo h1 {  
  font-weight: bold;  
  font-size: 25px;  
  color: #ff7062; /* color panetone 2019*/  
}
```

Aquests seran els estils per la barra del menú:

```
#menu{  
  width: 380px; /* la mida serà sempre la mateixa*/  
  float: left;  
}  
  
#menu ul li {  
  list-style: none;  
  float: left;  
  margin-left: 19px;  
  margin-right: 19px;  
}  
  
#menu ul li a {  
  color: black;  
}  
  
#menu ul li a:hover{  
  color: #ff7062;  
}
```

## Estils Slider

L'slider es el contenidor central que té un titular gran:

```
/*SLIDER*/

#slider {
    height: 300px;
    background: #ff7062;
}

#slider h2 {
    font-size: 40px;
    text-align: center;
    color: white;
    line-height: 290px;
    text-shadow: 0px 0px 6px black; /* ombra del text */
}
```

### Exemple de Progressive Web Apps

Serveis

## Estil Serveis

En la nostra pàgina web hi ha diversos subheaders els estils seràn per tots iguals

```
.subheader{
    font-size: 34px;
    color: #333;
    text-align: center;
    margin-top: 40px;
    margin-bottom: 30px;
    padding-bottom: 15px;
    border-bottom: 1px solid rgba(255, 112, 98, 0.3);
}
```

Serveis

Continuarem la maquetació per cada servei això el que farà serà:

```
.service {  
  width: 33%;  
  text-align: center;  
  float: left;  
  margin-top: 20px;  
  margin-bottom: 20px;  
}
```

Els icons seran tractats com a fonts:

```
.fa-grin-beam, .fa-grin-tears, .fa-grin-wink {  
  color: black;  
  font-size: 48px;  
  margin-bottom: 20px;  
}
```

Els subtítols i els paràgraf tindran aquests estils:

```
.service h3{  
  margin-bottom: 10px;  
}  
  
.service p {  
  color: #444;  
  margin: 5px;  
}
```

Aquest serà el resultat final:



## Estils Video

Els vídeos tindran els mateixos estils que els service.

```
.service, .video {  
  width: 33%;  
  text-align: center;  
  float: left;  
  margin-top: 20px;  
  margin-bottom: 20px;  
}
```

## Estils Footer

Aquests seran els estils pel footer:

```
/* FOOTER*/
{
  footer {
    background: #333;
    padding-top: 5px;
    padding-bottom: 40px;
    margin-top: 30px;
    text-align: center;
    color: white;
  }

  footer .subheader{
    color: white;
  }

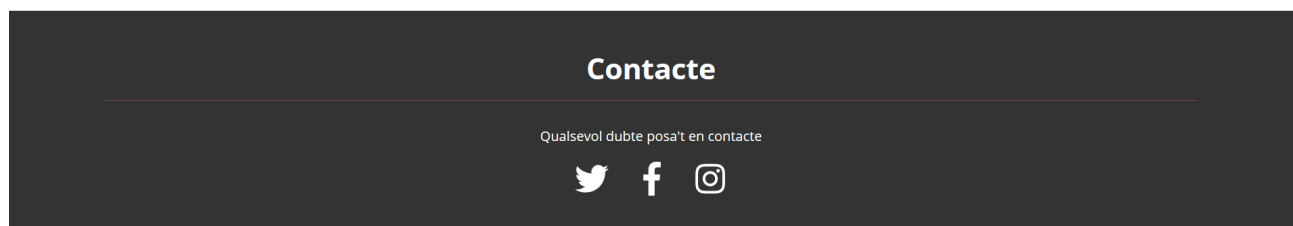
  footer ul{
    display: block;
    width: 220px;
    margin: 0px auto;
    list-style: none;
    margin-bottom: 20px;
    margin-top: 20px;
  }

  .fa-twitter, .fa-facebook-f, .fa-instagram{
    font-size: 40px;
    color: white;
  }

  .fa-twitter:hover, .fa-facebook-f:hover, .fa-instagram:hover{
    opacity: 0.8;
    transform: scale(1.1, 1.1);
  }

  footer ul li {
    float: left;
    margin-left: 20px;
    margin-right: 20px;
    transition: all 300ms;
  }
}
```

Aquest seria el resultat



# Maquetació responsive

El responsive design ens permet que una pàgina web es pugui veure en qualsevol dispositiu, actualment aquesta pàgina web només és visible per pantalles d'ordinador o superiors el que farem a partir d'aquest punt és que es pugui veure en tots els dispositius.

## Media Queries

Utilitzarem media query de CSS3 que ens permetrà limitar les fulles d'estil utilitzant característiques com l'alçada, l'amplada...

El primer que haurem de modificar és el document HTML perquè accepti les media queries:

```
<!-- Configuració del viewport per que accepti el responsive -->
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
```

Els media queries s'utilitzen amb la propietat **@media** i li indicarem el dispositiu l'aplicarem per defecte **all** però hi han altres opcions com **screen** (per pantalla d'ordinador) **print** (vista impressió) també li podem afegir expressions per exemple **@media all and (condició)** podem saber més sobre les media queries en aquest enllaç: [https://developer.mozilla.org/es/docs/CSS/Media\\_queries](https://developer.mozilla.org/es/docs/CSS/Media_queries)

Farem un exemple si al document styles.css escribim aquesta media query quan la pantalla tingui menys de 1000 px el body serà groc

```
@media all and (max-width: 1000px){
  body {
    background: yellow;
  }
}
```



Si escrivim aquesta altre media query quan la pantalla sigui superior a 1000px el body serà de color rosa.

```
@media all and (min-width: 1000px){  
  body {  
    background: pink;  
  }  
}
```

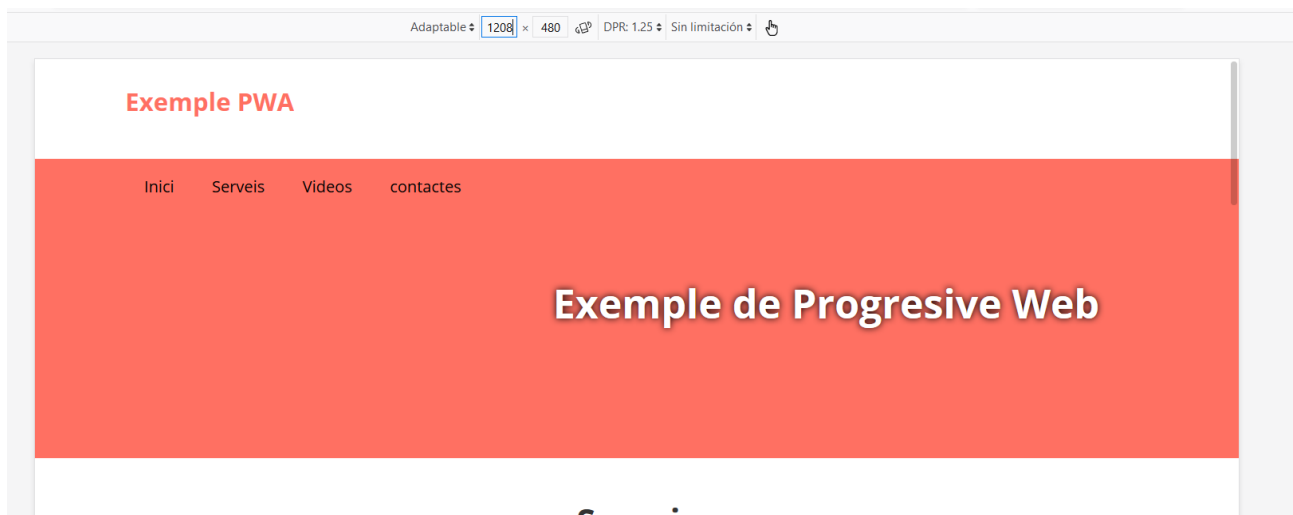


## Maquetació responsive capçalera

Per treballar en aquest apartat el que hem de fer és passar a vista d'administrador (clic botó esquerra → inspeccionar elemento) i a partir d'aquí començarem a buscar els punts de tall dels media queries.

El primer que farem serà posar un div amb la classe clearfix quan acabi el header al fitxer index.html i així netejar tots els flotats

podem veure que quan la pantalla arriba cap a 1808 px la capçalera ja no es comença a veure bé i els elements és desplacen.



Aquest és el nostre primer tall i farem una media query que quan sigui inferior a 1225px tingui els següents estils.

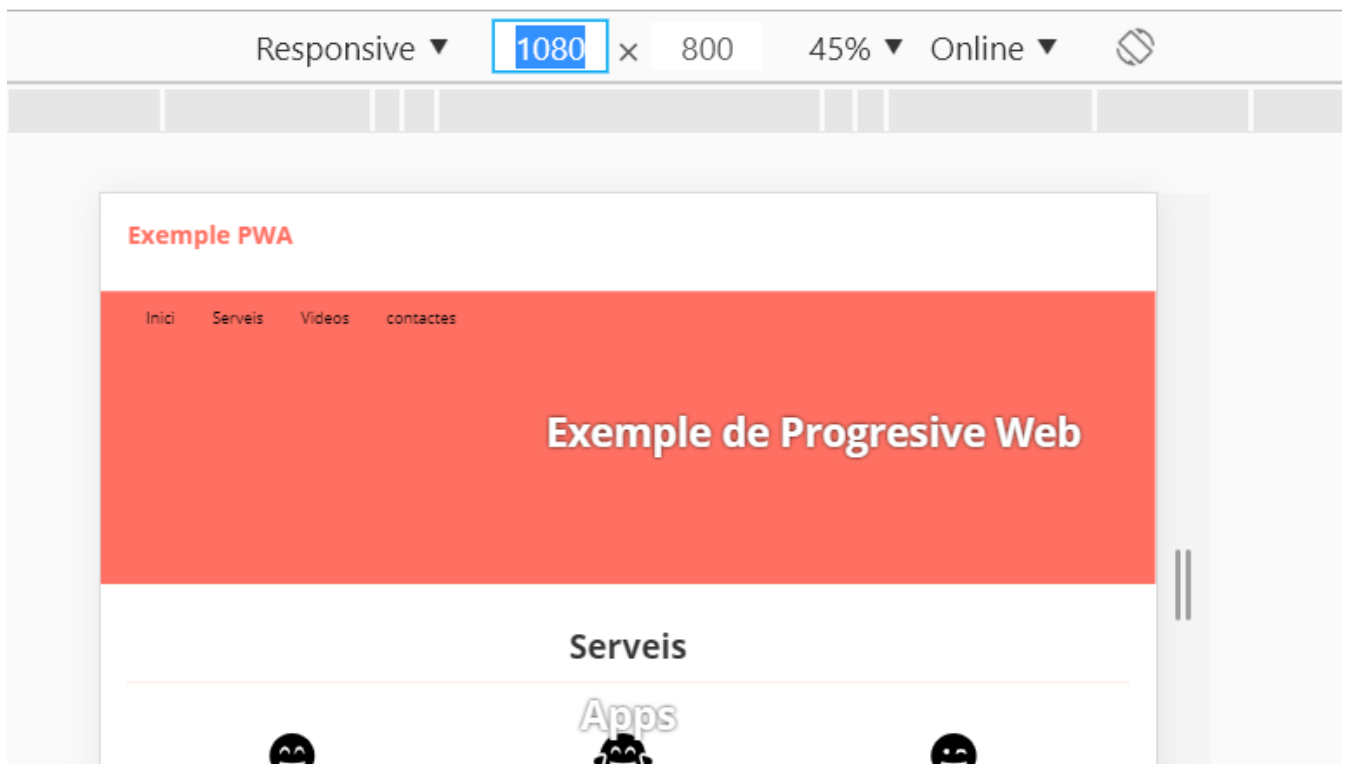
```
@media all and (max-width: 1225px){
  .container {
    width: 95%;
  }
}
```

D'aquesta manera quan la pantalla tingui una mida inferior a 1225px tindrà la següent vista.



El següent punt de tall el deixarem a 1085px ja que s'ens torna a moure la vista:





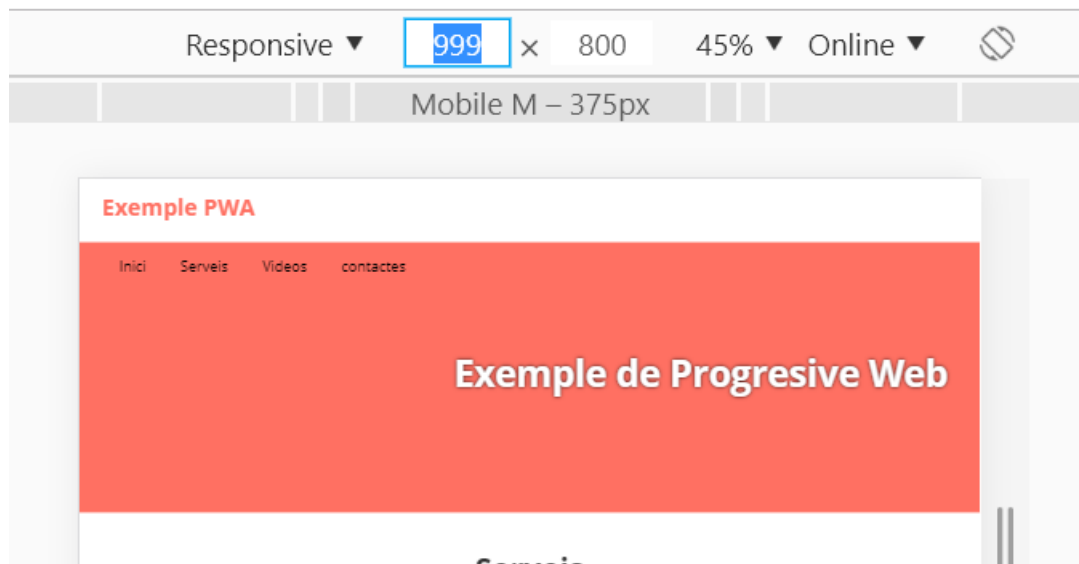
Com que s'ens mou el menú li posarem una mida al main-header un height de 70px i un line-height de 65px; i al logo una mida del 60% perquè no baixi el menú.

```
@media all and (max-width: 1085px){
  #main-header{
    height: 70px;
    line-height: 65px;
  }

  #logo{
    width: 60%;
  }
}
```



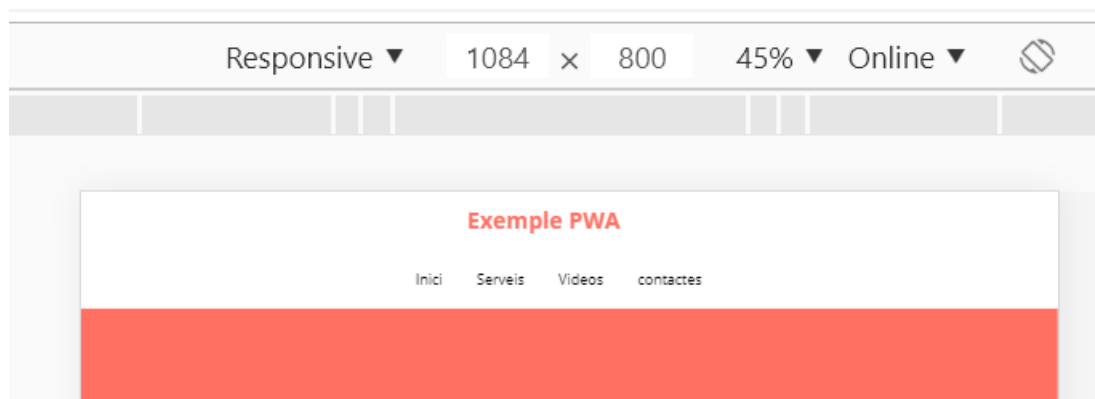
El següent punt de tall serà quan la pantalla faci 1000px ja que la web s'ens torna a descentrar



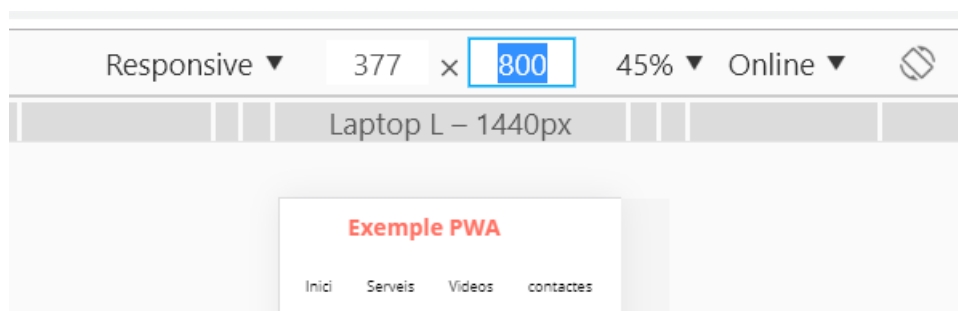
El que farem serà centrar la capçalera:

```
@media all and (max-width: 1085px){
  #main-header{
    height: 70px;
    line-height: 65px;
  }

  #logo{
    width: 60%;
  }
}
```



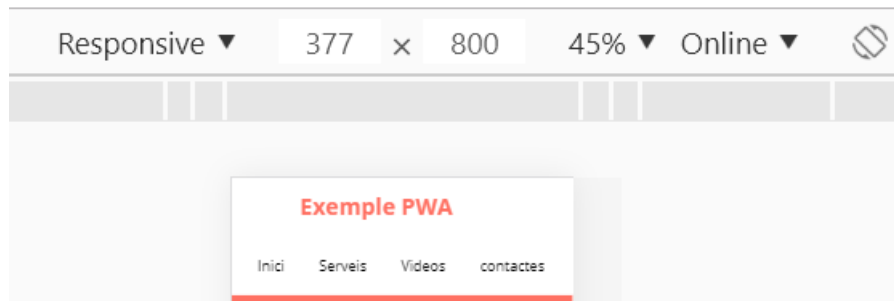
El següent punt de tall serà a 377 i el que farem serà reduir l'espai que hi ha entre els elements del menú.



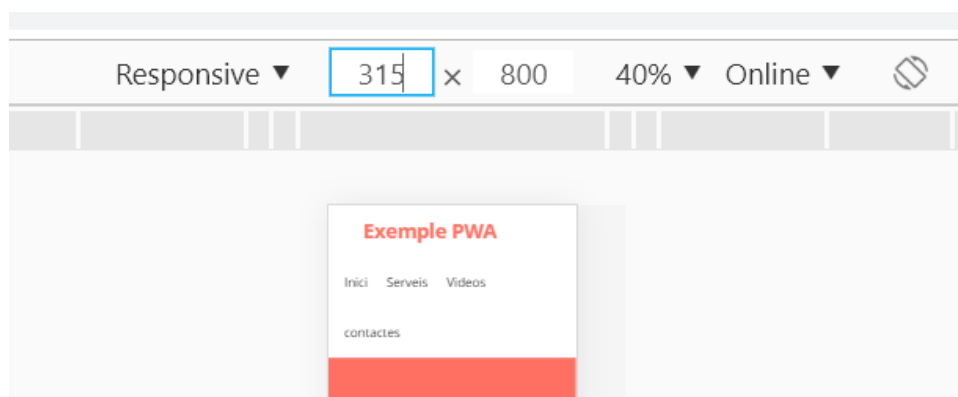
```

@media all and (max-width: 377px){
  #menu {
    width: 100%;
    text-align: center;
  }
  #menu ul li {
    margin-left: 4%;
    margin-right: 4%;
  }
}

```



El nostre últim tall pel que fa referència a la capçalera serà en el punt 345 que el menú es torna a descentrar. En aquest punt hauriem de fer un menú clicable, però per quatre elements li no val la pena i farem que es centrin un sota l'altre.

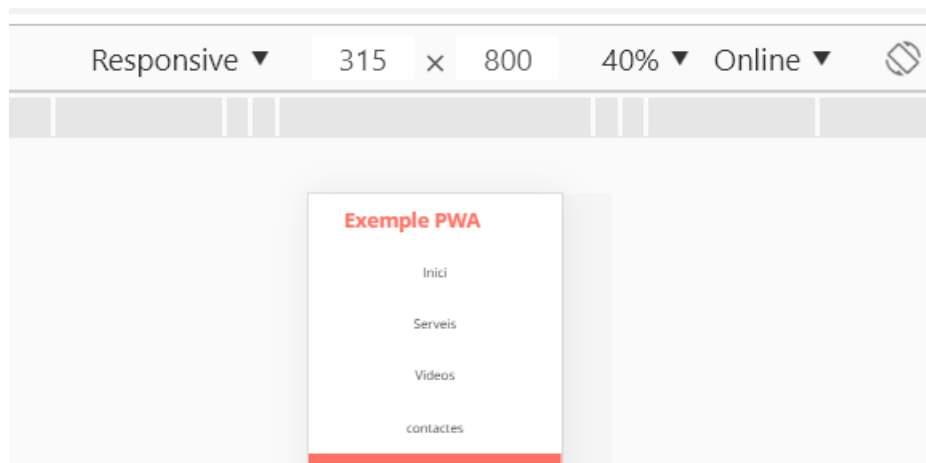


```

@media all and (max-width: 345px){
  #main-header {
    height: auto;
  }

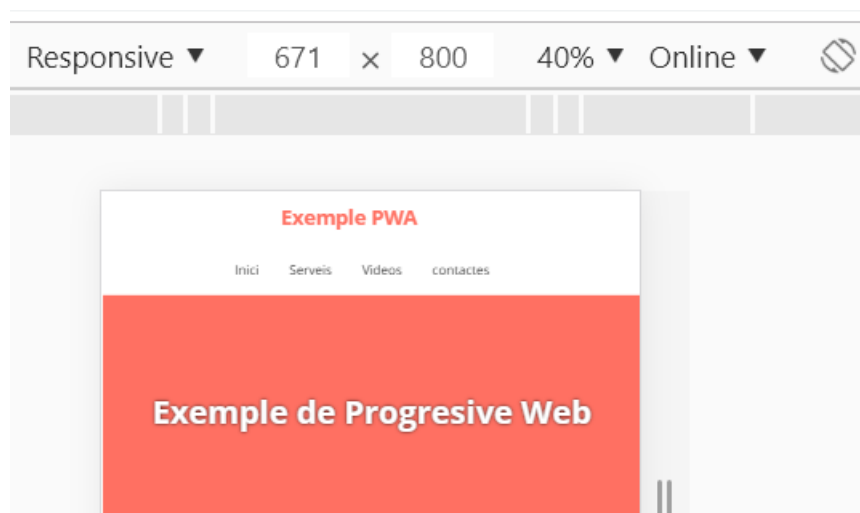
  #menu ul li {
    float: none;
  }
}

```



## Maquetació responsive slider

Continuarem fent les media queries per la part central a l'slider el nostre punt de tall serà al media query de la pantalla inferior a 999 i reduïrem el font-size



```
#slider {
  overflow: hidden;
  height: 150px;
}

#slider h2 {
  font-size: 30px;
  line-height: 30px;
  margin-top: 60px;
}
```

Quan arribem a 515 veiem que s'ens torna a descentrar el slider així que buscarem una media queri superior i també modificarem.



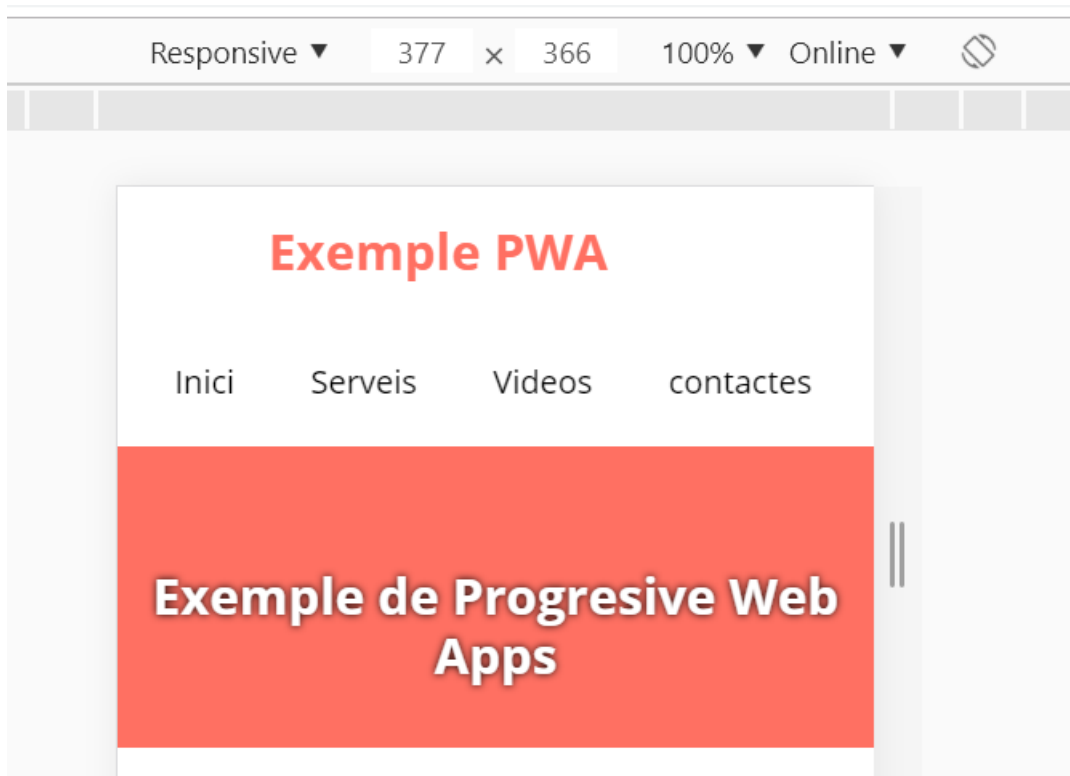
Quan la mida de la pantalla del dispositiu arribi a 460 el que farem serà una nova media queri que reduirà el line-height i el margin-top.



```
@media all and (max-width: 460px){
  #slider h2 {
    font-size: 25px;
    margin-top: 50px;
  }
}
```



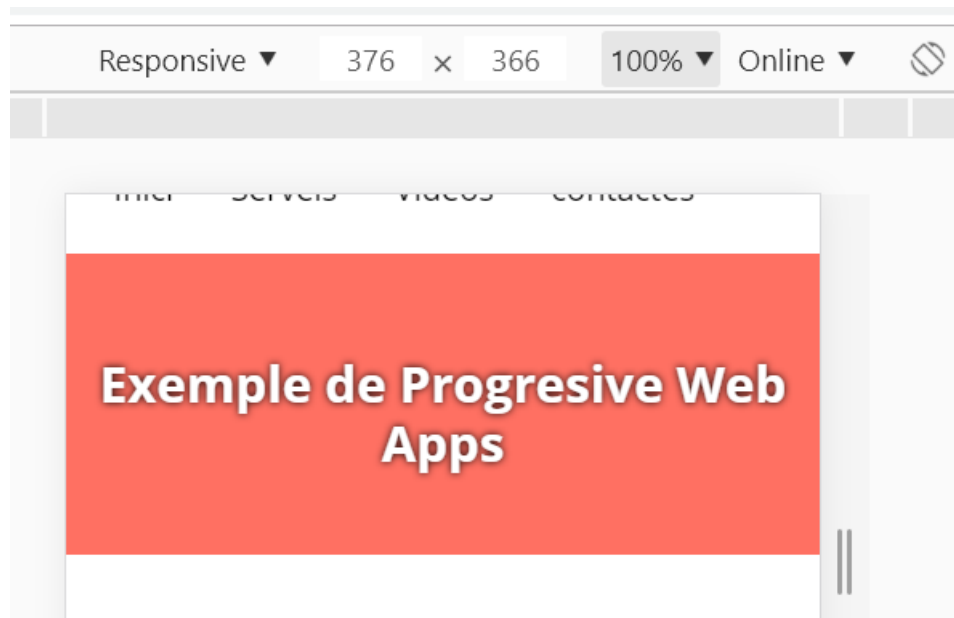
I en el següent media queri modificarem el line-height.



```

@media all and (max-width: 377px){
  #menu {
    width: 100%;
    text-align: center;
  }
  #menu ul li {
    margin-left: 4%;
    margin-right: 4%;
  }
  #slider h2 {
    margin-top: 50px;
  }
}

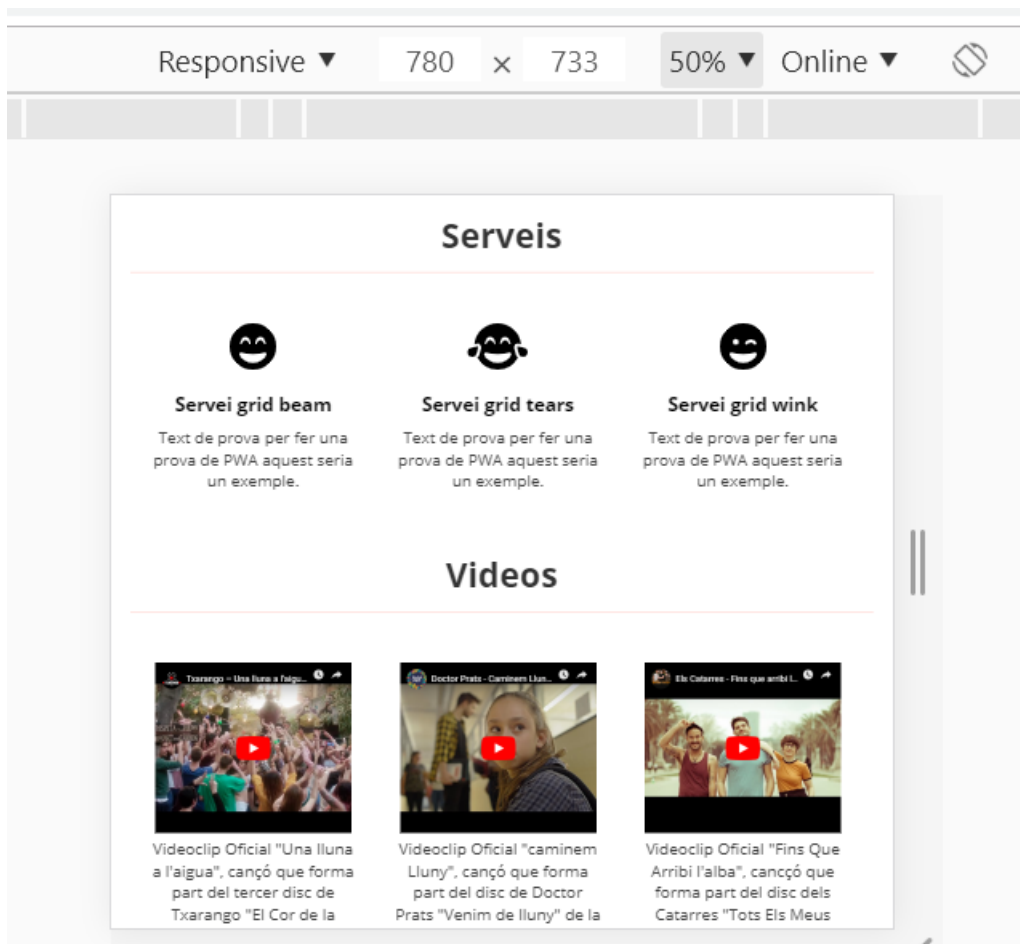
```



## Maquetació responsive serveis i video

Hem de tindre en compte que els serveis i els videos tenen moltes coses en comú així que la maquetació responsive la farem només en aquest punt.

Farem un media query nou a 780px per que la pàgina és vegi millor



```
@media all and (max-width: 780px){
  .service, .video {
    width: 70%;
    float: none;
    margin: 0px auto;
    margin-bottom: 40px;
  }

  .service p, .video p {
    margin-top: 10px;
  }

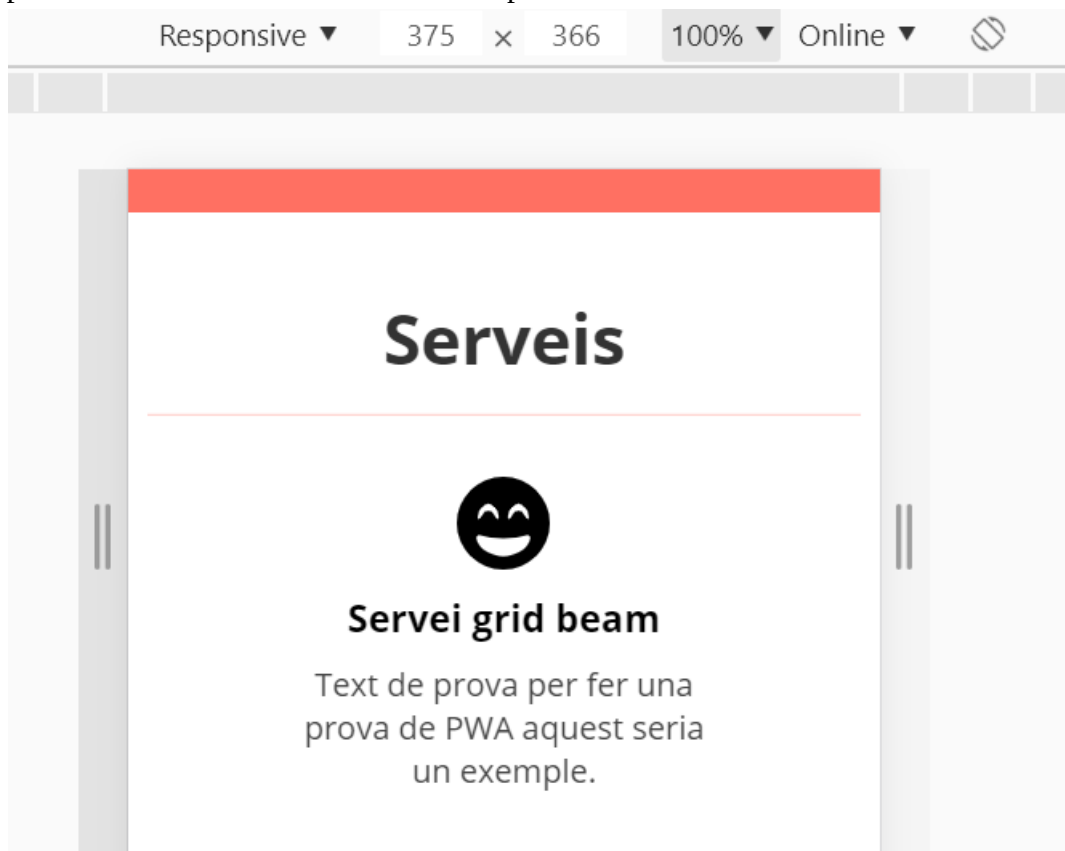
  .fa-grin-beam, .fa-grin-tears, .fa-grin-wink {
    margin-bottom: 10px;
  }

  .video iframe {
    width: 80%;
    height: 230px;
  }
}
```





Quan la pantalla arribi a 377 també farem unes petites modificacions:



```
@media all and (max-width: 377px){
  #menu {
    width: 100%;
    text-align: center;
  }
  #menu ul li {
```

```

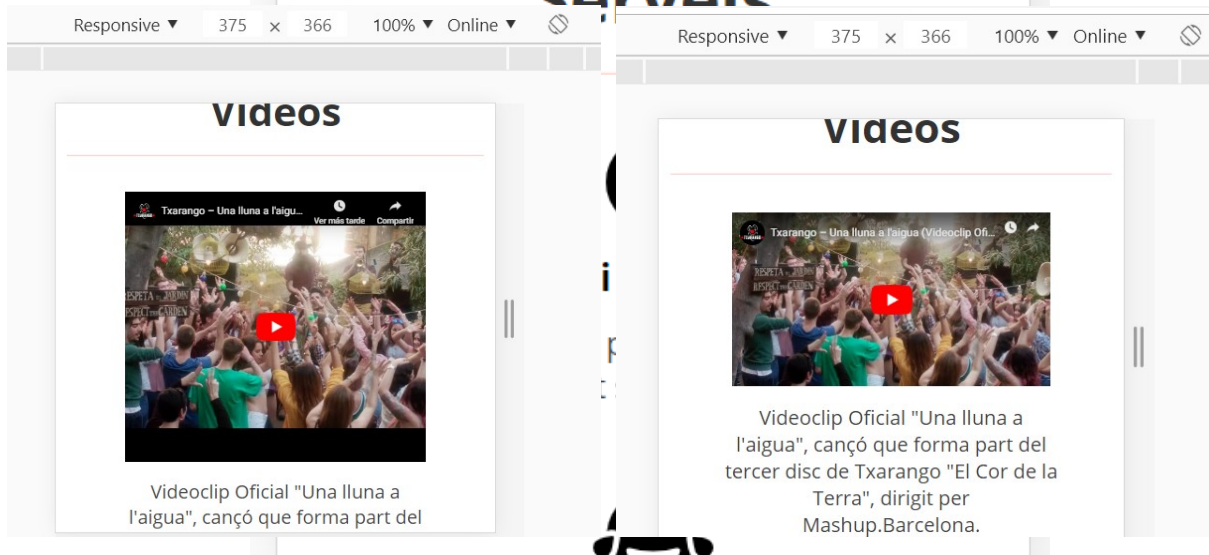
margin-left: 4%;
margin-right: 4%;
}

#slider h2 {
margin-top: 40px;
}

.service, .video {
width: 90%;
}
}

```

Modificarem els iframes



```

.video iframe {
height: 140px;
}

```

L'apartat de el footer no el modificarem dons ja s'adapta bé.

## Creant la PWA (Aplicació Web Progressiva)

Ens ha quedat pendent posar ancles al menú per que quan cliquem anem al lloc del menú que correspon:

```

<header id="main-header">
  <div class="container">
    <div id="logo">
      <h1>Exemple PWA</h1>
    </div> <!-- tanca logo -->
    <div id="menu">
      <ul>
        <li>
          <a href="#main-header">Inici</a>
        </li>
        <li>
          <a href="#services">Serveis</a>
        </li>
        <li>
          <a href="#videos">Videos</a>
        </li>
        <li>
          <a href="#footer">contactes</a>
        </li>
      </ul>
    </div> <!-- tanca menu -->
  </div> <!-- tanca container -->
</header> <!-- tanca header -->

```

També farem un scroll suau amb JQuery i per això afegirem el cdn

```

<!-- CDN JQuery -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

```

Crearem un arxiu js que anomenarem main.js i l'enllaçarem a index.html

```

<!-- main.js -->
<script type="text/javascript" src="js/main.js"></script>

```

Crearem la funció per fer l'scroll suau

```

1  $(document).ready(function(){
2      $("#menu a").click(function(e){
3          /* preventDefault evita la redirecció */
4          e.preventDefault();
5          |
6          $("html, body").animate({
7              scrollTop: $(this).attr('href').offset().top
8          });
9          |
10         return false;
11     });
12 });

```

## Etiquetes meta PWA

Configurarem les etiquetes de les metadades per que la PWA funcioni correctament:

## Etiqueta color aplicació

La primera etiqueta que configurarem serà la del color de l'aplicació.

```
<!-- metadada del color de l'aplicació -->  
<meta name="theme color" content="#ff7062" />
```

## Optimització mobile

La següent etiqueta indicarà que la nostra aplicació està optimitzada per mobil.

La primera etiqueta es creada des de versions antigües d'Internet Explorer al sistema Windows Mobile. Fa que el navegador adapti el contingut de la pàgina a la finestra del dispositiu. Permet que el navegador no faci cap canvi al layout de la pàgina que contingui aquesta META Tag.

La segona etiqueta informa al navegador que la pàgina està optimitzada per mòbils i que no s'ha d'escalar el seu contingut per mostrar-lo a la pantalla dels dispositius. Aquesta etiqueta va ser creada per els navegadors d'AvantGO, compatible amb dispositius BlackBerry, no s'ha popularitzat en moltes altres plataformes.

```
<!-- metadada indica aplicació optimitzada per mobil -->  
<meta name="MobileOptimized" content="width" />  
<meta name="HandheldFriendly" content="true" />
```

## Optimització Apple

La primera etiqueta indica si en content està a yes l'aplicació s'executa en mode de pantalla completa.

La segona etiqueta indica que si el contingut s'ha definit com a predeterminat, la barra d'estat és normal. Si està establert en negre, la barra d'estat té un fons negre. Si s'estableix en negre translúcid, la barra d'estat és negra i translúcida. Si s'estableix per defecte o en negre, el contingut web es mostra sota la barra d'estat. Si s'estableix en negre translúcid, el contingut web es mostra a tota la pantalla, parcialment obscurat per la barra d'estat. El valor per defecte és predeterminat.

```
<!-- Meta etiquetes PWA per Apple -->  
<meta name="apple-mobile-web-app-capable" content="yes">  
<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent">
```

## Icona apple

Configuració icones apple:

```
<meta name="apple-mobile-web-app-status-bar-style" content="black">  
<!-- icona PWA apple -->  
<link rel="apple-touch-icon" href="img/faviconPWA.png">  
<link rel="apple-touch-startup-image" href="img/faviconPWA.png">
```

## Manifest

El manifest és el fitxer de la configuració general de la PWA en format JSON.

```
<!-- Configuració General PWA -->  
<link rel="manifest" href="manifest.json" />
```

## manifest.json

<https://developer.mozilla.org/es/docs/Web/Manifest>

Aquest fitxer, que es un JSON és un dels documents més importants que té la PWA, ja que hi ha la configuració d'aquesta. Ha d'anar a l'arrel del projecte i obligatòriament s'ha d'anomenar manifest.json.

Aquest arxiu és una forma d'indicar als navegadors web, i en especial als dispositius mòbils, com mostrar la teva App on s'especifiquen diverses propietats amb els seus valors.

### Propietat name

Aquesta propietat defineix el nom de la pàgina pot ser el mateix que el title o un de personalitzat.

```
"name": "PWA",
```

### Propietat short\_name

Aquesta propietat defineix el nom que apareix a l'icona de l'app. No necessàriament ha de ser igual que la propietat name tot i que en l'exemple coincideix.

```
"short_name": "PWA",
```

### Propietat description

Aquesta propietat defineix el què fa la nostra aplicació en el nostre cas és una mostra d'una PWA.

```
"description": "Exemple d'Progressive Web Application",
```

### Propietat background\_color

Color que tindrà de fons esperat per la PWA. Aquest valor és igual que el del full CSS però els navegadors poden utilitzar el color de fons d'una adreça quan el manifest està disponible abans que el full d'estils estigui carregat. Això crea una transició suau entre el llançament de l'aplicació web i la carrega del contingut de la web.

```
"background_color": "#fff",
```

### Propietat theme\_color

Aquesta propietat defineix el color principal de l'aplicació.

```
"theme_color": "#ff7062",
```

## Propietat orientation

Aquesta propietat defineix l'orientació de la PWA pot ser "landscape" per que el dispositiu estigui en horitzontal o "portrait" per que el dispositiu estigui en vertical.

```
"orientation": "portrait",
```

## Propietat display

Aquesta propietat defineix com es mostrara l'aplicació web al nostre mòbil. És pot mostrar en tipus "browser" i veurem la barra del navegador o "standalone" i ens amagarà el navegador

```
"display": "standalone",
```

## Propietat start\_url

Defineix la pàgina que es veurà al principi quan es carregui l'aplicació. I no necessàriament ha de ser index.html el paràmetre que es passa indica que és una web progressiva.

```
"start_url": "./index.html?utm_source=web_app_manifest",
```

## Propietat scope

Defineix l'àmbit de navegació en el contexte de l'aplicació web. Això basicament restringeix quines pàgines es poden veure quan s'aplica el manifest. Si l'usuari navega fora l'àmbit de l'aplicació continua com una web normal.

```
"scope": "./",
```

## Propietat lang

Defineix el codi que utilitzarà la PWA

```
"lang": "ca-CA",
```

## Propietat icons configuracio icones

Per carregar les icones de l'aplicació primer de tot les hem de tindre i les hem de tindre en els següents formats :

1024px X 1024px, 512px X 512px, 384px X 384px, 256px X 256px, 192px X 192, 128px X 128px, 96px X 96px, 64px X 64px, 32px X 32px i 16px X 16px les podem escalar amb GIMP i per guardar-les a exportar como.

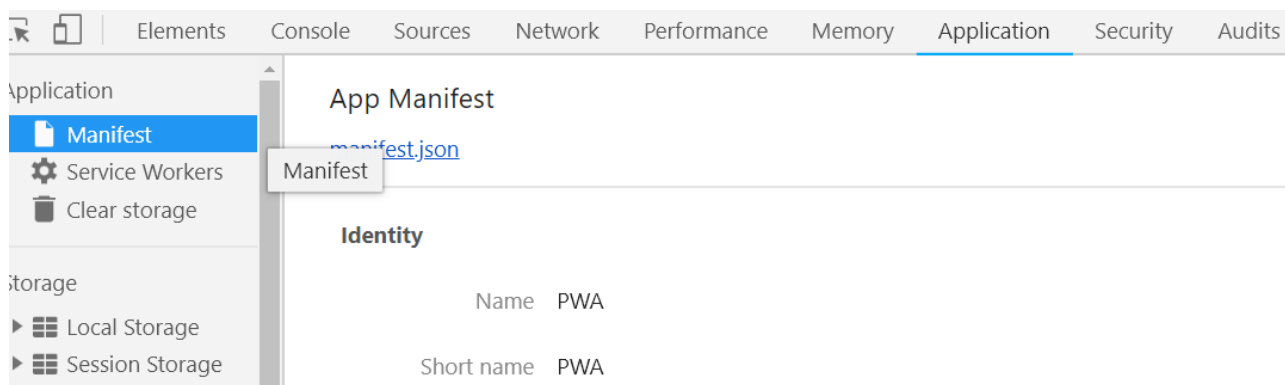
Carregarem la propietat icons que és un array en format JSON

```

lang : ca=CA ,
"icons": [
  {
    "src": "./img/faviconPWA-1024.png",
    "sizes": "1024x1024",
    "type": "image/png"
  },
  {
    "src": "./img/faviconPWA-512.png",
    "sizes": "512x512",
    "type": "image/png"
  },
  {
    "src": "./img/faviconPWA-384.png",
    "sizes": "384x384",
    "type": "image/png"
  },
  {
    "src": "./img/faviconPWA-256.png",
    "sizes": "256x256",
    "type": "image/png"
  },
  {
    "src": "./img/faviconPWA-192.png",
    "sizes": "192x192",
    "type": "image/png"
  },
  {
    "src": "./img/faviconPWA-128.png",
    "sizes": "128x128",
    "type": "image/png"
  },
  {
    "src": "./img/faviconPWA-96.png",
    "sizes": "96x96",
    "type": "image/png"
  },
  {
    "src": "./img/faviconPWA-64.png",
    "sizes": "64x64",
    "type": "image/png"
  },
  {
    "src": "./img/faviconPWA-32.png",
    "sizes": "32x32",
    "type": "image/png"
  },
  {
    "src": "./img/faviconPWA-16.png",
    "sizes": "16x16",
    "type": "image/png"
  }
]

```

Per comprovar que el manifest.json carrega correctament podem anar a chrome i a applications podem veure que carrega el manifest.json.



## Service Worker

[https://developer.mozilla.org/es/docs/Web/API/Service\\_Worker\\_API](https://developer.mozilla.org/es/docs/Web/API/Service_Worker_API)

<https://desarrolloweb.com/articulos/service-workers.html>

Els service workers tenen sentit en molts escenaris, però bàsicament ens permeten mantenir un lloc web en segon pla. Bàsicament constitueixen en un script (JavaScript) que continua la seva execució, independentment de si la pàgina web està o no oberta.

Els Service Workers permeten instal·lar una Web en un dispositiu i simular el comportament a la app sense connexió a internet.

Els Service Workers no poden accedir directament al DOM, el Service Worker es comunica amb les pàgines que controla a través de l'interfaç PostMessage.

Amb un Service Worker pots controlar com les peticions de la xarxa.

Els Service Workers poden implementar diverses sistemes de cacheig.

Poden persistir la informació a través del indexedDB.

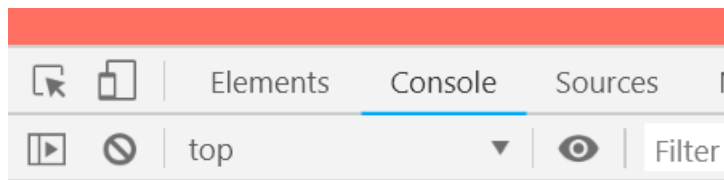
Es necessari que el lloc web s'entregui mitjançant un protocol segur per que el Service Worker pugui ser registrat.

## Creant el Service Worker

Anirem al fitxer main.js, que haurem traslladat a l'arrel del projecte, i escriurem el següent codi preguntant si podem fer servir el service worker al navegador. Com a prova farem un console.log() per veure el resultat.

```
//Configuració service worker
if('serviceWorker' in navigator){
  console.log('Es pot utilitzar el service worker');
}else{
  console.log('no es pot utilitzar el service worker');
}
```



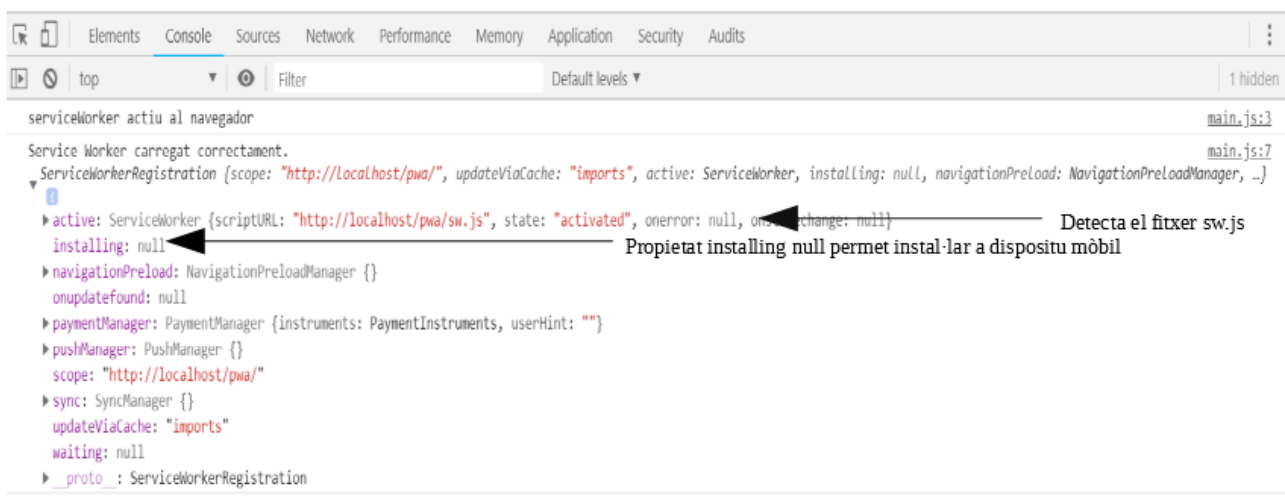


Es pot utilitzar el service worker

Com que es molt complexe el service worker crearem un nou fitxer a l'arrel del projecte que anomenarem sw.js aquí és on guardarem el service worker però a main.js carregarem el sw.js

```
1 //Configuració service worker
2 if('serviceWorker' in navigator){
3   console.log('serviceWorker actiu al navegador');
4
5   //carrega del service worker amb un registre
6   navigator.serviceWorker.register('./sw.js')
7     .then(res => console.log('Service Worker carregat correctament.', res))//retorna una promesa
8     .catch(err => console.log('Service Worker no s\'ha pogut registrar', err)); //en cas error
9 }else{
10  console.log('no es pot utilitzar el serviceWorker al navegador.');
```

Refrescarem i veurem que el service worker està enllaçat correctament.



## Event install

Aquesta part de codi l'escriurem a sw.js el primer que farem serà assignar un nom i una versió de cache.

```
1 //Assignem nom i versió a la cache
2 const CACHE_NAME = 'v1_cache_cristina_roige_prova_pwa';
```

Crearem un array per guardar els arxius que volguem que es guardin a la cache:

- Fitxers del directori actual.
- Fitxer fulla d'estils css/styles.css
- Les imatges que hi ha als fitxers img

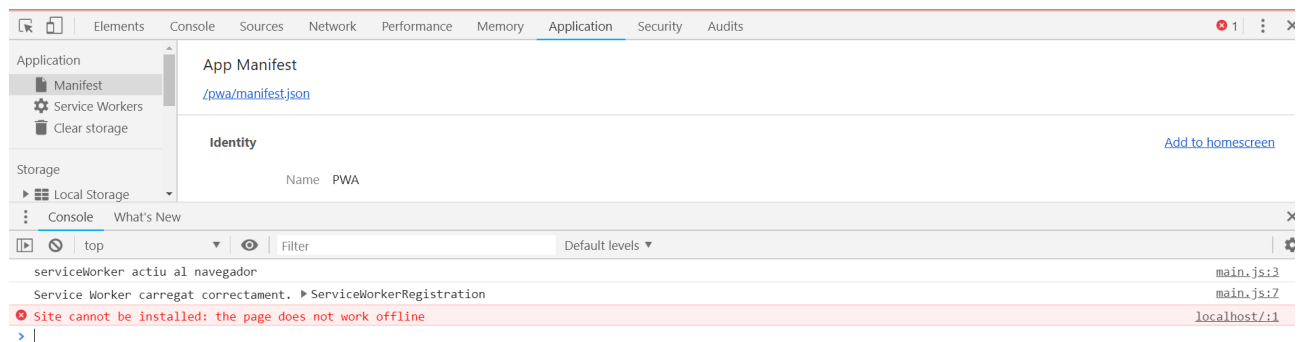
- Fitxer js/scroll.js
- Fitxer js/fontawesome/all.js

```
//fitxers que es guardaran a la cache es un array d'arxius
var urlsToCache = [
  './', //fitxers del directori actual
  './css/styles.css', // fulla d'estils
  './img/faviconPWA-1024.png',
  './img/faviconPWA-512.png',
  './img/faviconPWA-384.png',
  './img/faviconPWA-256.png',
  './img/faviconPWA-192.png',
  './img/faviconPWA-128.png',
  './img/faviconPWA-96.png',
  './img/faviconPWA-64.png',
  './img/faviconPWA-32.png',
  './img/faviconPWA-16.png',
  './img/faviconPWA.ico',
  './img/faviconPWA.png',
  './js/scroll.js', //fitxers js
  './js/fontawesome/all.js'
];
```

Crearem els events que té el `service worker` que solen ser instal·lació activació i fetch (El que aconseguix la informació des d'internet).

Crearem el event install.

Si ara anem al inspector d'elements de chrome i mirem a application veurem que hi ha un link que diu add to homescreen i si hi cliquem quan els events estiguin tot fets s'instal·larà l'aplicació, però de moment no s'instal·larà i donarà el missatge "*Site cannot be installed: the page does not work offline*".



## Event Activate

Farem igual que a l'event install i utilitzarem `self.addEventListener('activate', e => {})` farà que l'aplicació funcioni offline.

El que fa el mètode activate és recorre els elements de la cache i esborra els que no hi són o no es necessiten.

```

//Event activate
// utilitzarem self.addEventListener amb una funció de callback

self.addEventListener('activate', e => {
  //crearem una constant que anomenarem cacheWhiteList que contindrà
  //tots els elements que hi haurà a la cache
  const cacheWhiteList = [CACHE_NAME];

  e.waitUntil(
    //cridarem el objecte caches i al mètode keys
    //el mètode keys recull el que hi ha a la cache i és una promesa
    caches.keys()
      .then(cacheNames => {
        return Promise.all(
          //recorrem els elements de la cache amb map
          cacheNames.map(cacheName => {
            //buscarem dins de la cache l'element que estem recorrent
            //si no hi ha l'element programàticament a la cache guardada
            //s'eliminarà
            if (cacheWhiteList.indexOf(cacheName) === -1) {
              return caches.delete(cacheName)
            }
          })
        );
      })
    //funció callback sense paràmetre que activa la cache actual al dispositiu
    //i l'app funcioni sense connexió.
    .then(() => self.clients.claim());
  );
});

```

## Event fetch

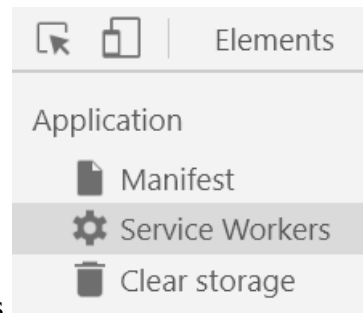
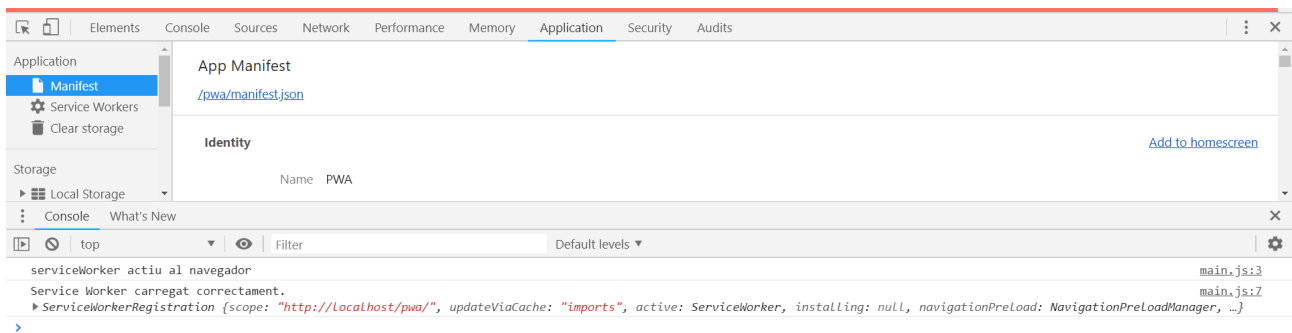
Quan sol·licitem una url l'event fetch comprovarà si la informació està en cache i si està en cache retorna els recursos de la cache i si no els troba a la cache els sol·licita a la web.

```

//Event fetch
self.addEventListener('fetch', e => {
  //reponem amb dades de la cache o dades que recuperem del servidor
  e.respondWith(
    //Si troba e.request es una promesa i utilitzem then
    caches.match(e.request)
      .then(res => {
        //si fa match i dona true retorna dades de la cache
        if (res) {
          return res
        }
        //en qualsevol altre cas petició ajax a la url original
        return fetch(e.request)
      })
  );
});

```

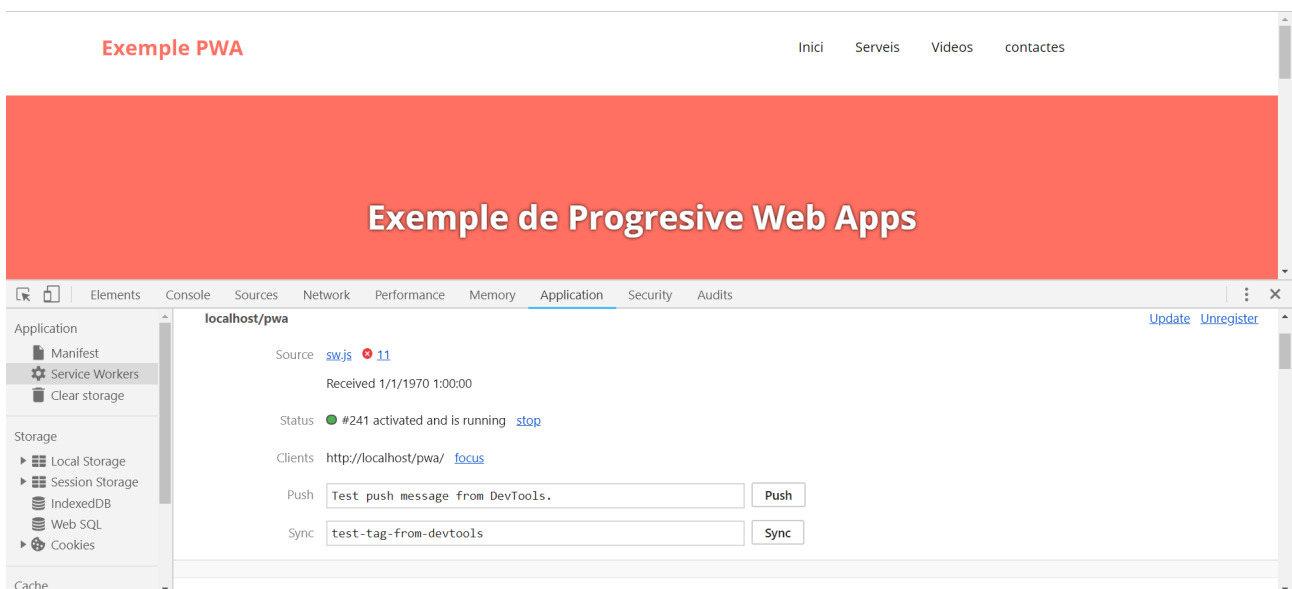
Refrescarem la nostra pàgina webi veurem que el service worker funciona correctament



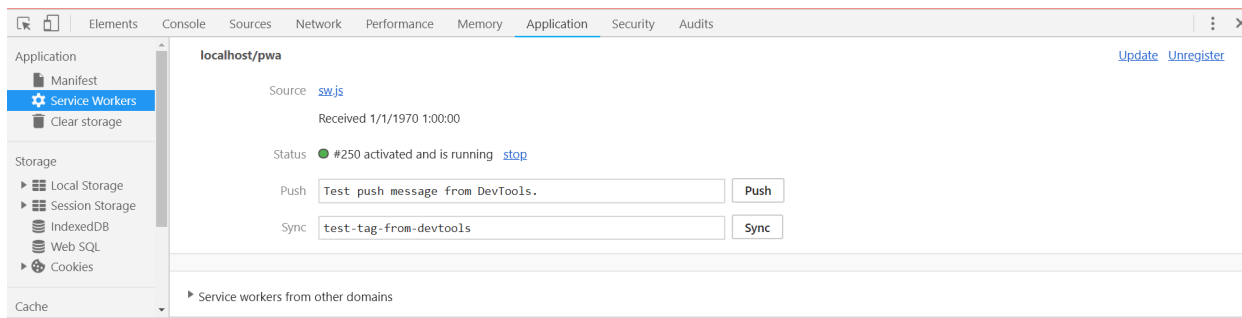
Això ho podem veure clicant a Service Workers

Veiem que el service worker està actiu

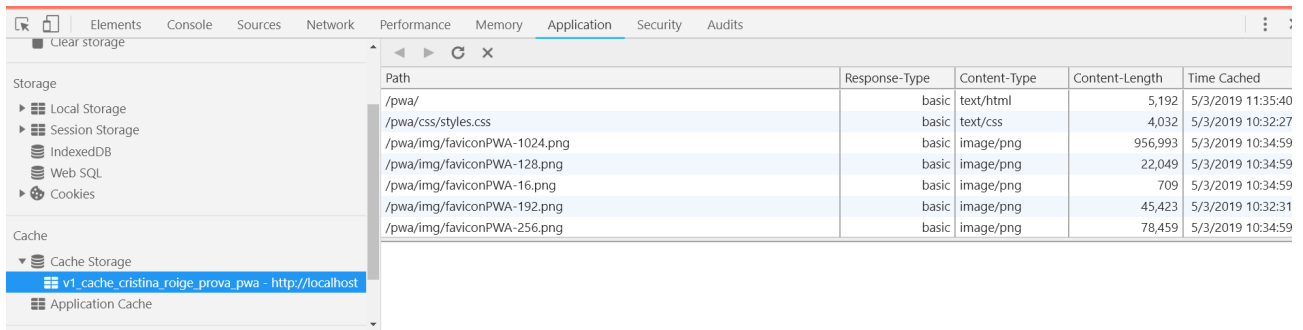
Status ● #241 activated and is running [stop](#)



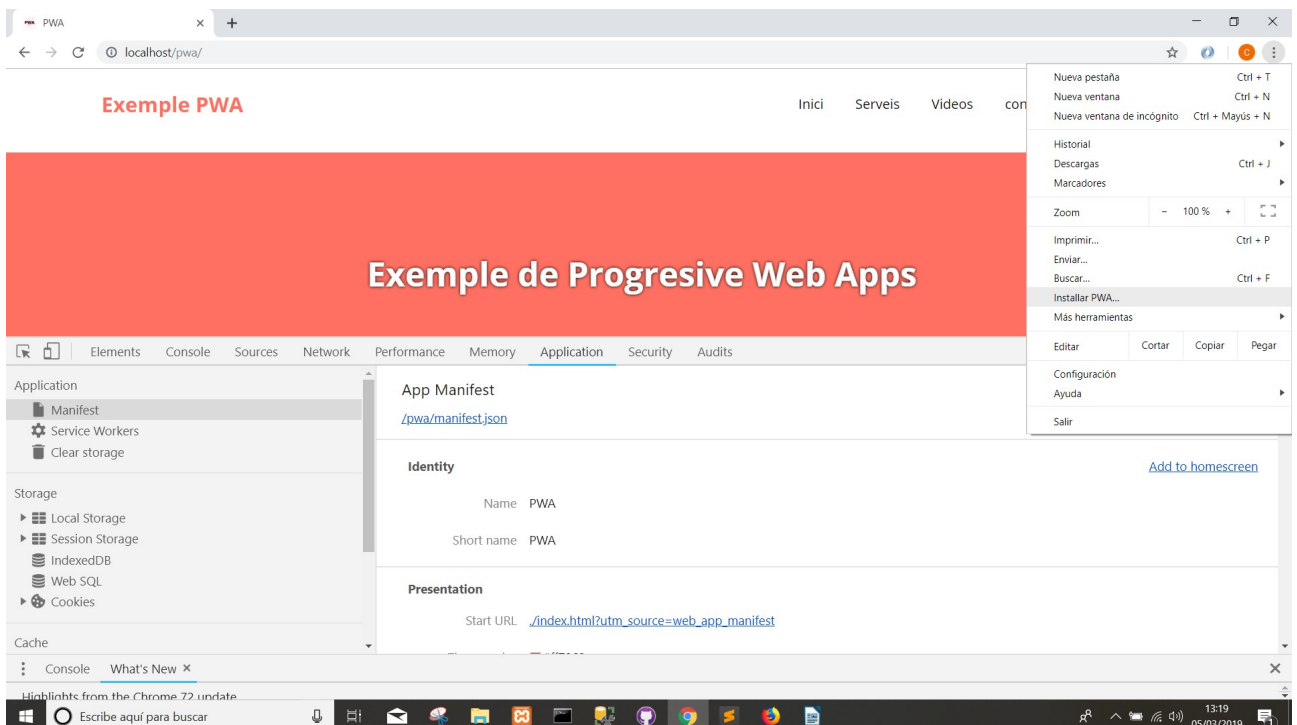
Recordatori: ctrl + f5 instal·lem el service worker.



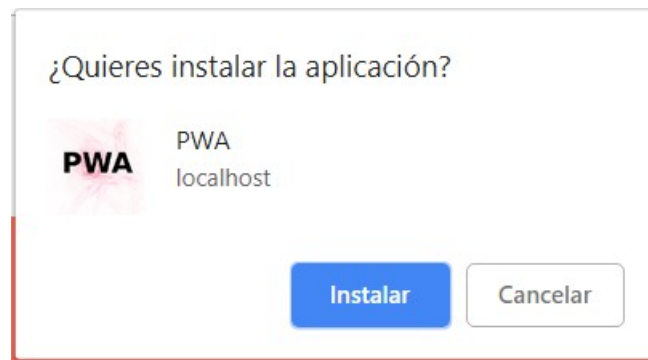
Si anem a cache storage i despleguem podem veure tots els arxius que hi ha en cache.



Si anem al manifest hi ha el botó add to homescreen i per defecte el navegador també dona l'opció d'instal·lar la pwa.



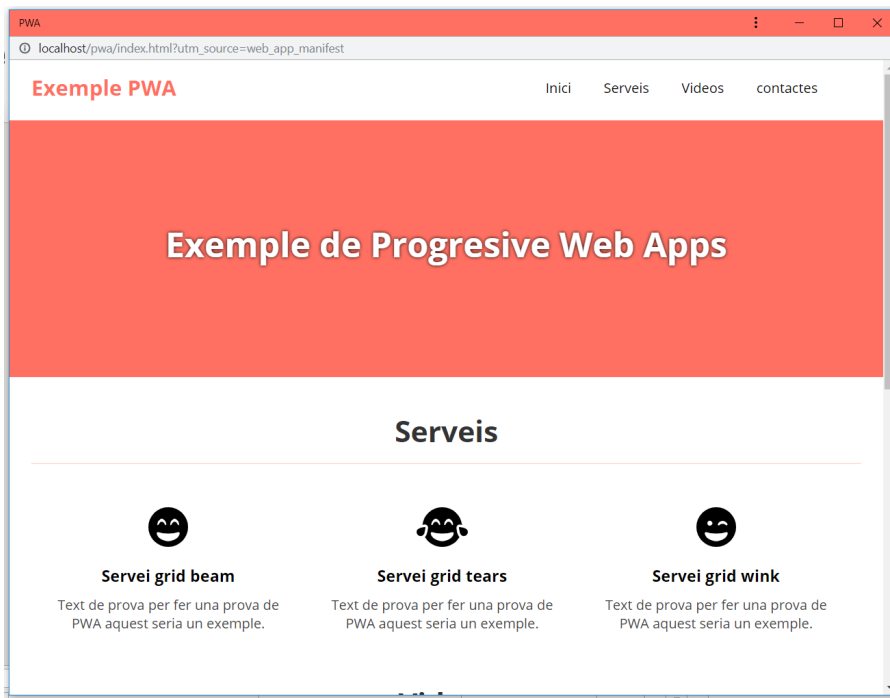
Si cliquem podem veure que ens surt una finestra que ens diu si volem instal·lar la pwa



Si cliquem que si veurem que al nostre escriptori hi haurà la icona de la nostra PWA



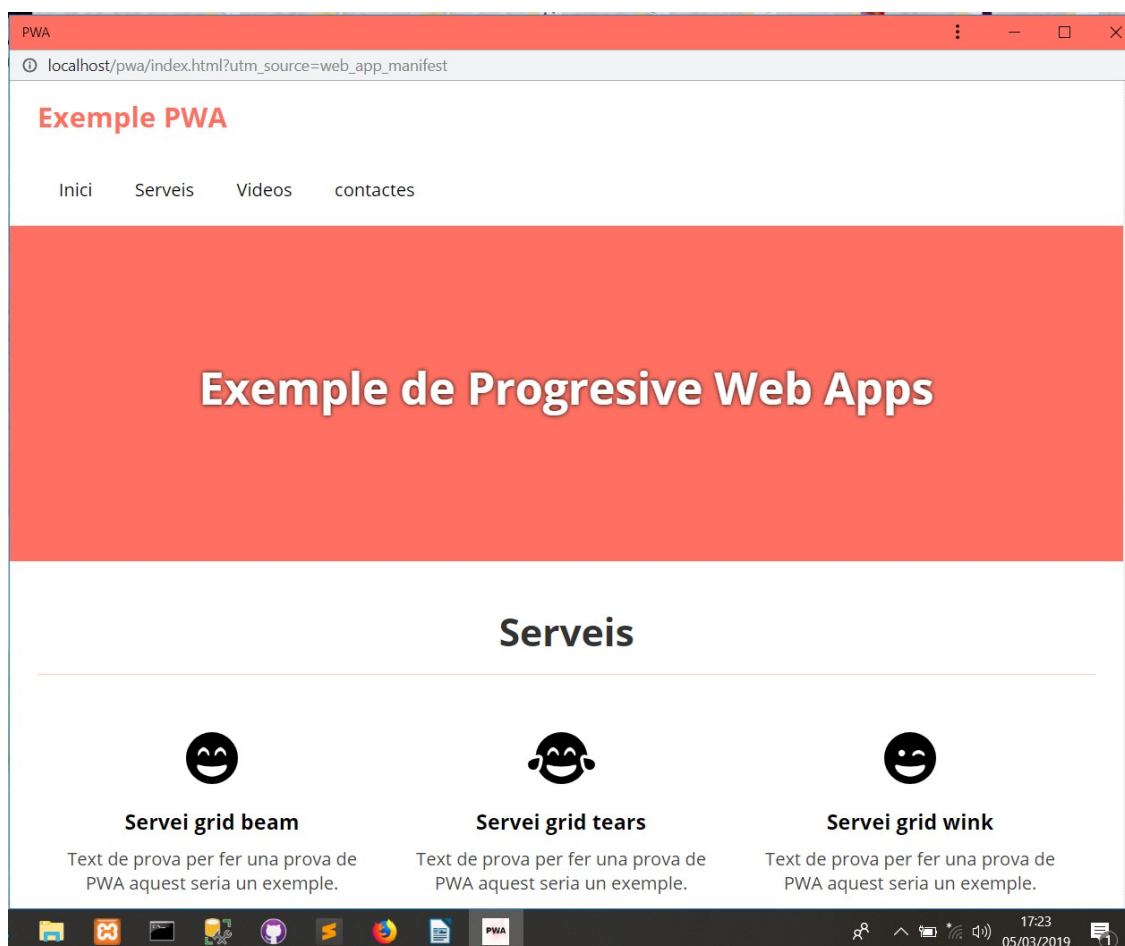
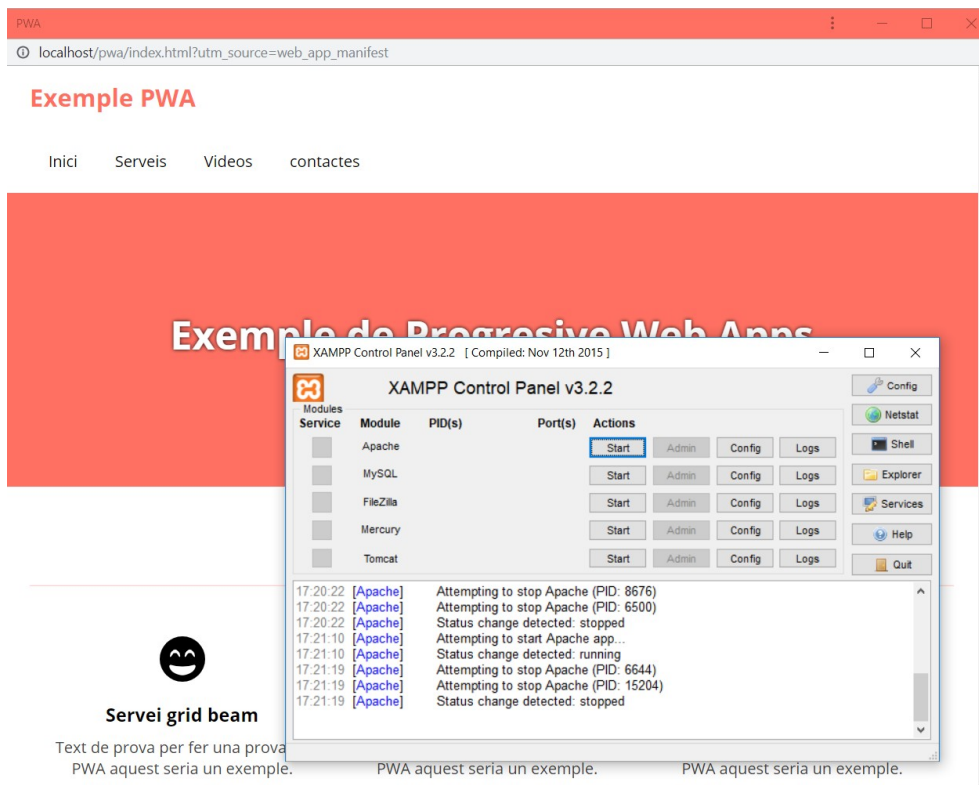
Si la obrim podem veure que es comporta com una aplicació d'escriptori.



Si canviem la mida podem veure que respon al responsive



Si paro el meu servidor podem veure que funciona correctament (menys els iframes)

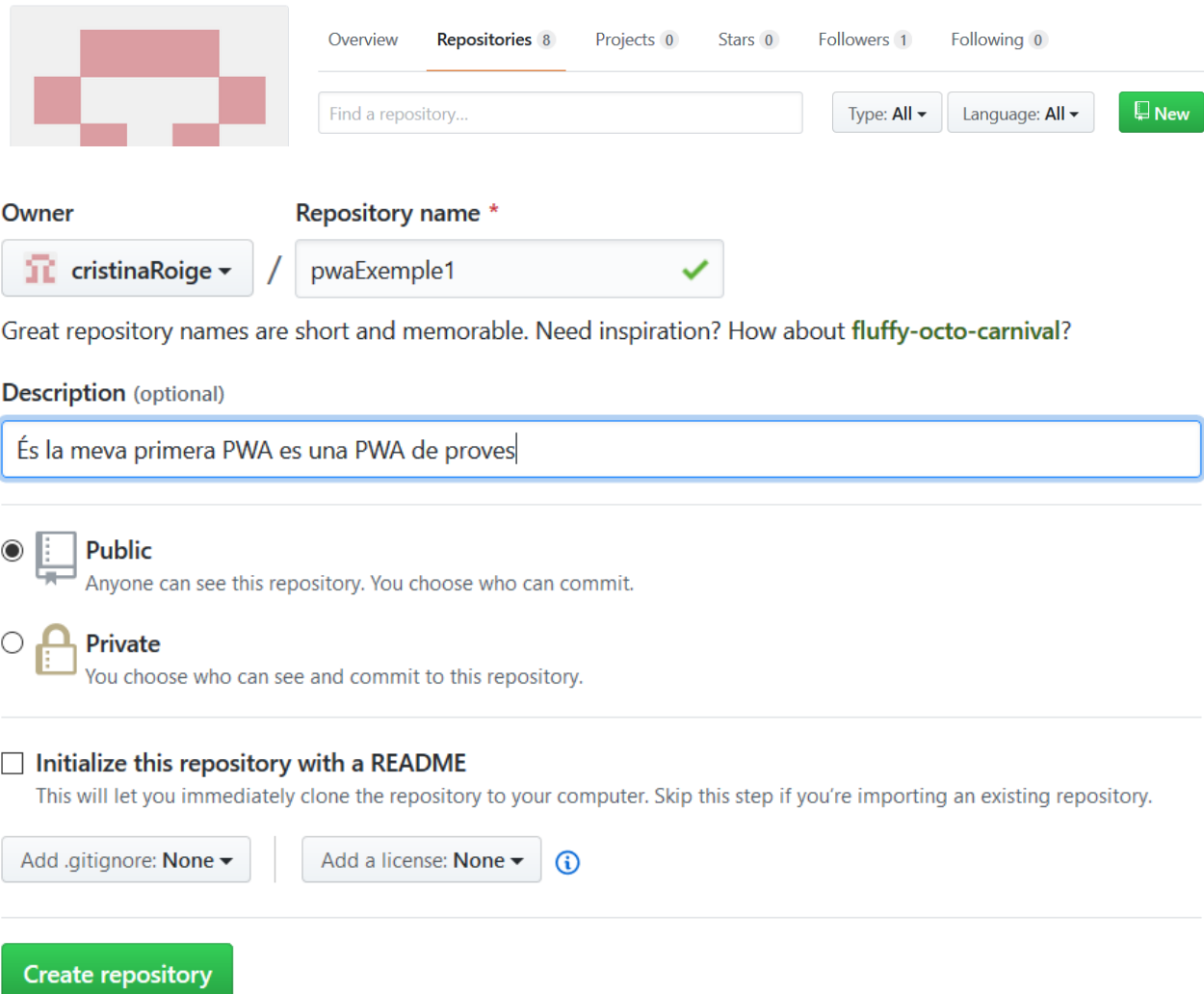




# Publicar PWA servidor web

Aprofitarem el hosting que ens dona GitHub que és un hosting gratuït tot i que no dona servei de backend. És un hosting amb un protocol segur que les pwa es necessiten.

Entrarem a GitHub i crearem un nou repositori i clicarem create repository:



Overview Repositories 8 Projects 0 Stars 0 Followers 1 Following 0

Find a repository... Type: All Language: All New

Owner Repository name \*

cristinaRoige / pwaExemple1 ✓

Great repository names are short and memorable. Need inspiration? How about [fluffy-octo-carnival?](#)

Description (optional)

És la meva primera PWA es una PWA de proves

☒ Public Anyone can see this repository. You choose who can commit.

☐ Private You choose who can see and commit to this repository.

☐ Initialize this repository with a README This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None Add a license: None ⓘ

Create repository

Veurem les instruccions per pujar el repository:

## ...or create a new repository on the command line

```
echo "# pwaExemple1" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/cristinaRoige/pwaExemple1.git
git push -u origin master
```

Obrem la consola de comandes i anirem al directori on tenim la pwa

```
Microsoft Windows [Versión 10.0.17134.590]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Cristina>cd ..\..\

C:\>cd xampp\htdocs\pwa

C:\xampp\htdocs\pwa>
```

Llançarem la comanda git init per iniciar el repositori local

```
C:\xampp\htdocs\pwa>git init
Initialized empty Git repository in C:/xampp/htdocs/pwa/.git/

C:\xampp\htdocs\pwa>
```

Si fem git status per veure el que hi ha dins podem veure el següent:

```
C:\xampp\htdocs\pwa>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    Recursos/
    css/
    img/
    index.html
    js/
    main.js
    manifest.json
    sw.js

nothing added to commit but untracked files present (use "git add" to track)
```

Llancem la comanda git add .

```
C:\xampp\htdocs\pwa>git add .
```

i git commit -m "Primer commit"

```
C:\xampp\htdocs\pwa>git commit -m "Primer commit"
[master (root-commit) f18cc6a] Primer commit
```

git status

```
C:\xampp\htdocs\pwa>git status
On branch master
nothing to commit, working tree clean
```

Si volem veure el commit que hem fet:

```
C:\xampp\htdocs\pwa>git log --oneline
f18cc6a (HEAD -> master) Primer commit
```

Hem pujat codi a una rama master en el nostre cas:

```
C:\xampp\htdocs\pwa>git branch
* master
```

Volem que el nostre repositori sigui una web de GitHub Pages crearem una pàgina específica:

```
C:\xampp\htdocs\pwa>git branch gh-pages
```

El que pugem a aquesta branca serà una pàgina web de GitHub per canviar-nos a aquesta branca

```
C:\xampp\htdocs\pwa>git checkout gh-pages
Switched to branch 'gh-pages'
M    Recursos/.~lock.apunts.odt#
M    Recursos/apunts.odt
D    Recursos/img.zip
D    Recursos/img/img/1.png
D    Recursos/img/img/2.png
D    Recursos/img/img/3.png
D    Recursos/img/img/4.png
D    Recursos/img/img/5.png
D    Recursos/img/img/6.png
D    Recursos/img/img/facebook.png
D    Recursos/img/img/favicon-1024.png
D    Recursos/img/img/favicon-128.png
D    Recursos/img/img/favicon-16.png
D    Recursos/img/img/favicon-192.png
D    Recursos/img/img/favicon-256.png
D    Recursos/img/img/favicon-32.png
D    Recursos/img/img/favicon-384.png
D    Recursos/img/img/favicon-512.png
D    Recursos/img/img/favicon-64.png
D    Recursos/img/img/favicon-96.png
D    Recursos/img/img/favicon.png
D    Recursos/img/img/instagram.png
D    Recursos/img/img/twitter.png
```

Comprovem a on som:

```

C:\xampp\htdocs\pwa>git status
On branch gh-pages
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Recursos/.~lock.apunts.odt#
        modified:   Recursos/apunts.odt
        deleted:    Recursos/img.zip
        deleted:    Recursos/img/img/1.png
        deleted:    Recursos/img/img/2.png
        deleted:    Recursos/img/img/3.png
        deleted:    Recursos/img/img/4.png
        deleted:    Recursos/img/img/5.png
        deleted:    Recursos/img/img/6.png
        deleted:    Recursos/img/img/facebook.png
        deleted:    Recursos/img/img/favicon-1024.png
        deleted:    Recursos/img/img/favicon-128.png
        deleted:    Recursos/img/img/favicon-16.png
        deleted:    Recursos/img/img/favicon-192.png
        deleted:    Recursos/img/img/favicon-256.png
        deleted:    Recursos/img/img/favicon-32.png
        deleted:    Recursos/img/img/favicon-384.png
        deleted:    Recursos/img/img/favicon-512.png
        deleted:    Recursos/img/img/favicon-64.png
        deleted:    Recursos/img/img/favicon-96.png
        deleted:    Recursos/img/img/favicon.png
        deleted:    Recursos/img/img/instagram.png
        deleted:    Recursos/img/img/twitter.png

no changes added to commit (use "git add" and/or "git commit -a")
C:\xampp\htdocs\pwa>

```

Pujem el codi a la branca

```
C:\xampp\htdocs\pwa>git remote add origin https://adrecarepositoriRemot
```

```
C:\xampp\htdocs\pwa>git remote add origin https://github.com/cristinaRoige/pwaExemple1.git
```

```

C:\xampp\htdocs\pwa>git push -u origin gh-pages
Enumerating objects: 66, done.
Counting objects: 100% (66/66), done.
Delta compression using up to 8 threads
Compressing objects: 100% (62/62), done.
Writing objects: 100% (66/66), 10.05 MiB | 58.00 KiB/s, done.
Total 66 (delta 10), reused 0 (delta 0)
remote: Resolving deltas: 100% (10/10), done.
To https://github.com/cristinaRoige/pwaExemple1.git
 * [new branch]      gh-pages -> gh-pages
Branch 'gh-pages' set up to track remote branch 'gh-pages' from 'origin'.

```

Ara si anem a <https://cristinaroige.github.io/pwaExemple1/> podem veure la pàgina web ja pujada i que funciona correctament i la podem fins i tot instal·lar.

