

# Guia d'ús de programació: *plugin* WordPress per a Mosaic

UOC

## Índex

1. Instal·lar l'extensió
2. Incorporar el codi
  - 2.1. Incorporar altres recursos
  - 2.2. Mostrar el nostre programa
    - 2.2.1. Fer servir l'*id* d'un element HTML
    - 2.2.2. Fer servir un *shortcode*
3. Accedir a les dades sobre el lloc web

Autoria: Marc Padró

PID\_00267116



CC BY-NC-ND  
Primera edició: setembre 2019  
Autoria: Marc Padró  
Llicència CC BY-NC-ND d'aquesta edició, FUOC, 2019  
Av. Tibidabo, 39-43, 08035 Barcelona  
Realització editorial: FUOC

Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement- NoComercial-SenseObraDerivada (BY-NC-ND) v.3.0 Espanya de Creative Commons. Podeu copiar-los, distribuir-los i transmetre'ls públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), no en feu un ús comercial i no en feu obra derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.ca>

# p5.js Embed: guia d'ús

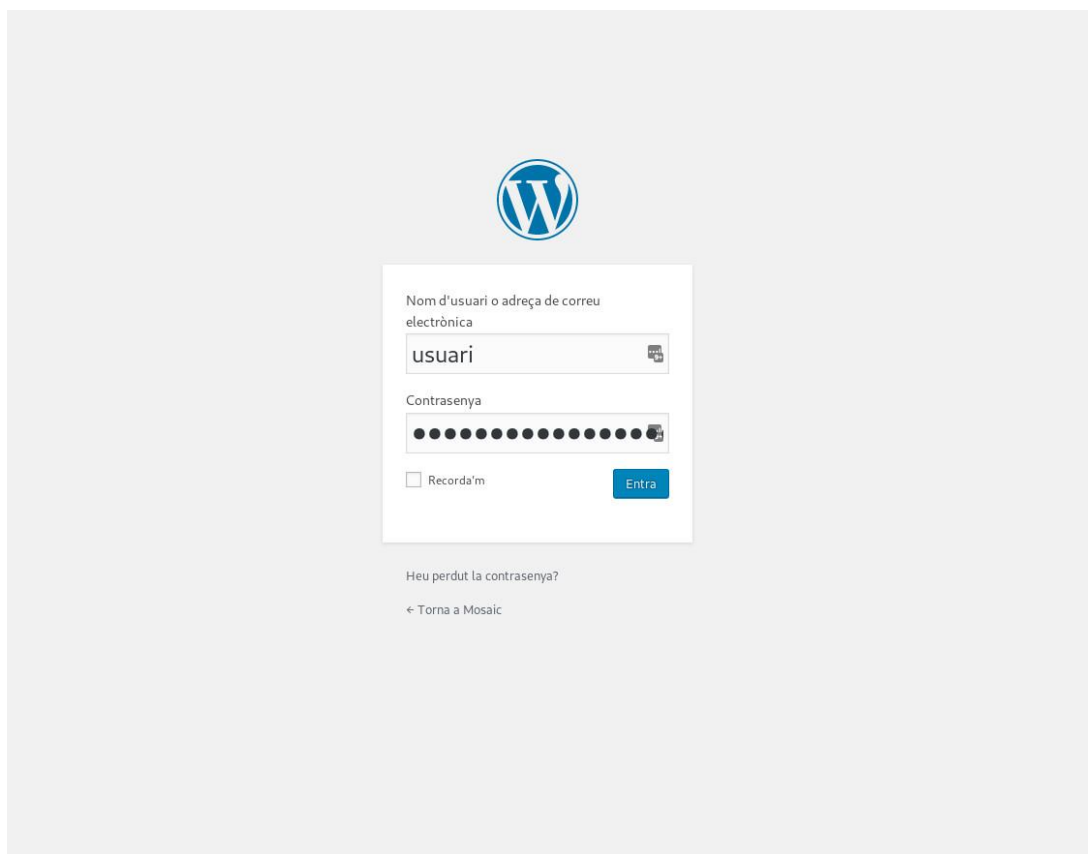
'p5.js Embed' és una extensió per a WordPress que us permetrà incorporar codi de p5.js en un lloc web que funcioni amb aquesta plataforma. A més, aquesta extensió permet accedir des del codi p5.js a dades sobre el lloc web i la pàgina que s'està consultant.

## 1. Instal·lar l'extensió

Per instal·lar l'extensió necessitareu l'arxiu *p5js-embed.zip* disponible amb els recursos de l'assignatura.

Primer de tot cal accedir a l'administrador del WordPress on vulguem instal·lar-la. Per fer-ho afegim */admin* al final de l'adreça principal del lloc: per exemple, si l'adreça del nostre lloc és *exemple.com*, ens dirigirem a *exemple.com/admin*.

Figura 1. Pantalla d'inici de sessió a WordPress



Un cop haguem entrat les nostres credencials (necessitarem un usuari amb permís d'administrador), accedirem al tauler de control de WordPress. Cal que ens dirigim a l'apartat 'Extensions' (o *Plugins* en anglès), que és al menú situat a l'esquerra de la pàgina. Posant-hi el punter a sobre, accedim al subapartat 'Afegeix' (figura 3).

## Nota

En aquesta guia fem servir termes de WordPress de la seva versió en català. Si la vostra instal·lació és en un altre idioma, els textos canviaran, però generalment seran traduccions directes dels que fem servir aquí. Si ho voleu, podeu canviar l'idioma de la vostra instal·lació a l'apartat 'Opcions' del menú.

Figura 2. Pantalla inicial del tauler de control de WordPress

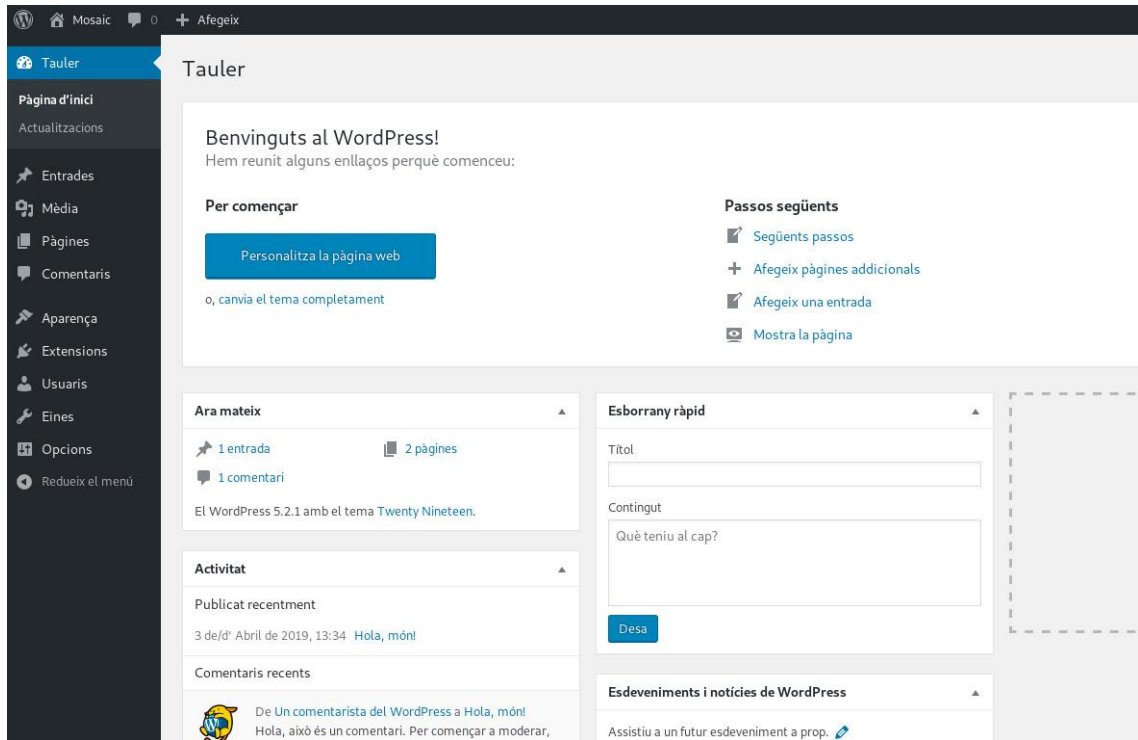
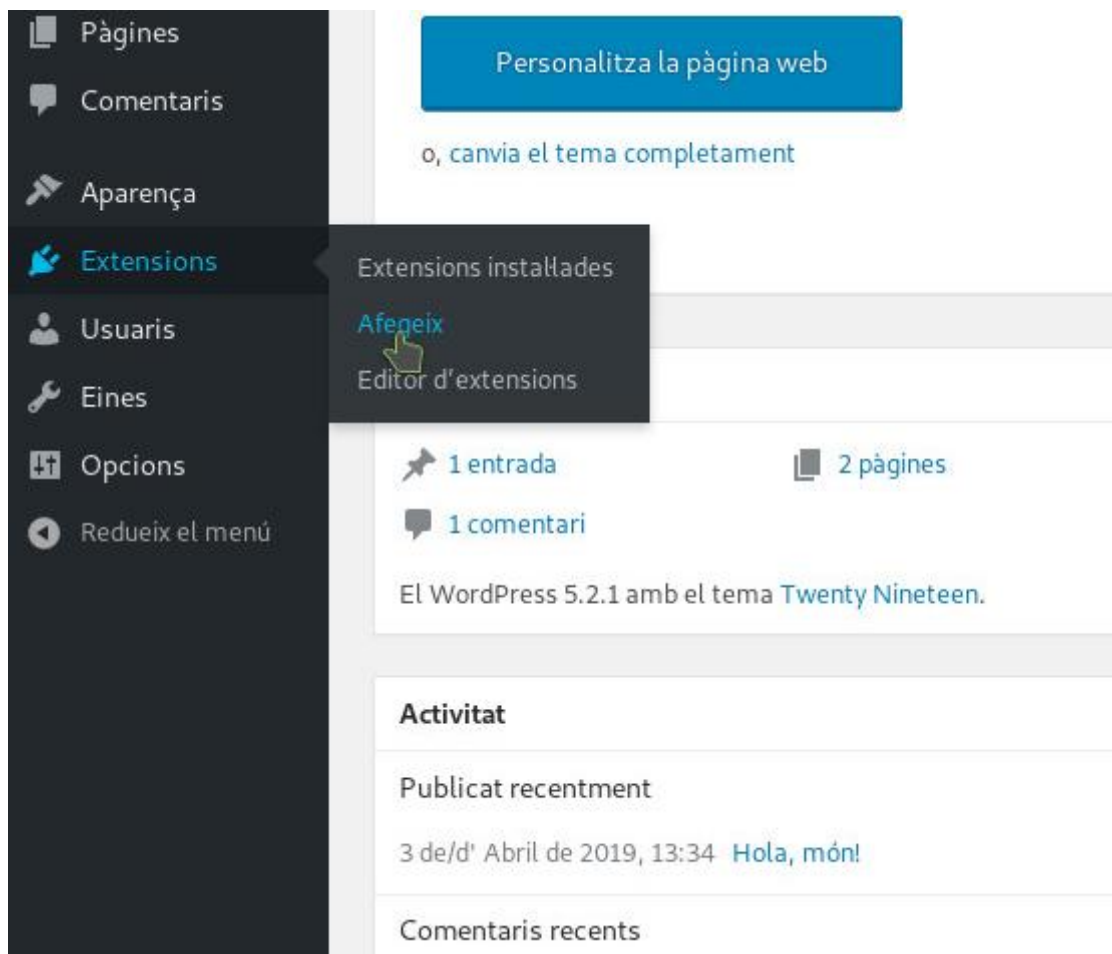


Figura 3. Apartat al menú per afegir extensions de WordPress



Un cop a dins, al principi de la pàgina, al costat del títol, hi trobarem el botó 'Penja l'extensió' (figura 4). Hi clicarem.

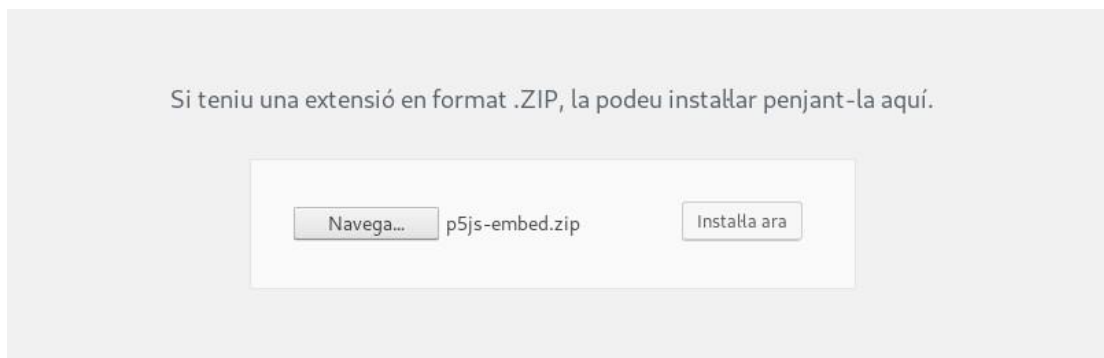
Figura 4. Botó per penjar una extensió mitjançant un arxiu del nostre ordinador



Se'ns obre llavors un diàleg que ens permet pujar un arxiu ZIP (figura 5). Fem clic al botó per buscar l'arxiu que hem de pujar (el text del botó pot variar segons el navegador) i seleccionem l'arxiu *p5js-embed.zip*.

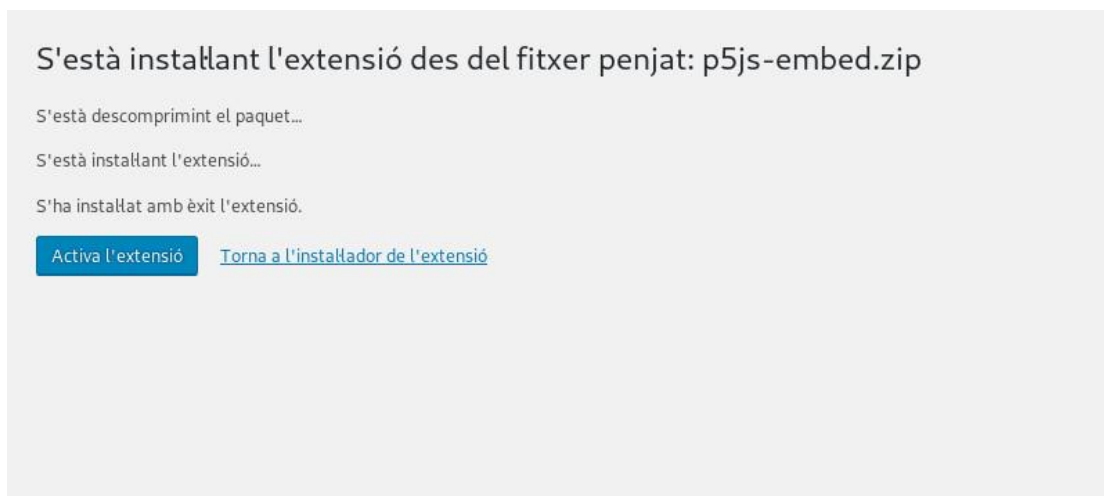
Un cop seleccionat l'arxiu, cliquem al botó 'Instal·la ara'.

Figura 5. Interfície per seleccionar i pujar l'arxiu de l'extensió



Llavors ens apareixerà un text que ens informa del progrés de la instal·lació. Un cop finalitzada, fem clic a 'Activa l'extensió'.

Figura 6. Text de progrés de la instal·lació de l'extensió



## 2. Incorporar el codi

Un cop instal·lada l'extensió, per incorporar el codi de p5.js ens hem de dirigir a l'apartat 'p5.js Embed' dins de la pestanya 'Opcions' del menú (figura 7).

Figura 7. Apartat dins el menú per accedir a la configuració de 'p5.js Embed'

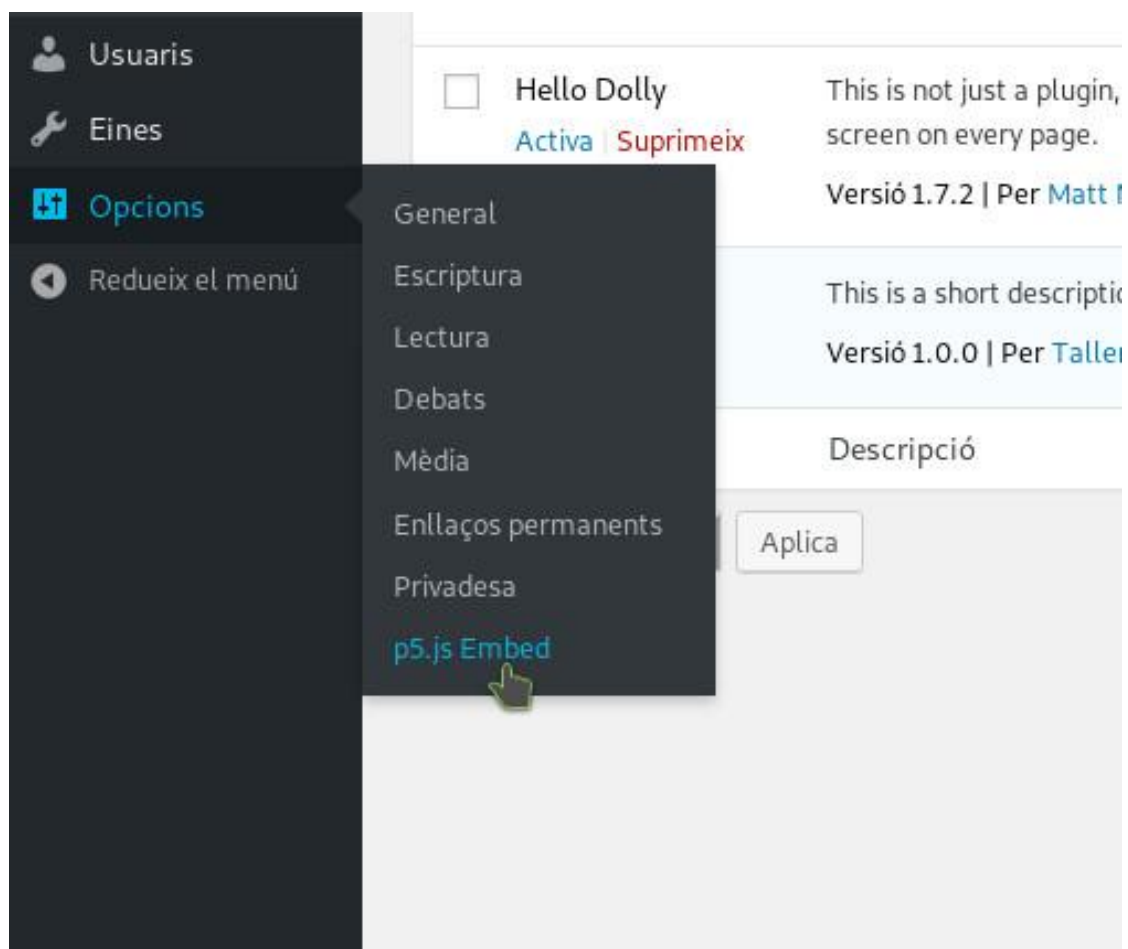


Figura 8. Opcions de configuració de 'p5.js Embed'

The screenshot shows the 'p5.js Embed' configuration interface. It includes the following elements:

- Embed p5.js code**: A section header.
- Container id**: A text input field.
- Sketch width**: A dropdown menu with '300' selected.
- Sketch height**: A dropdown menu with '300' selected.
- Addons**: Two checkboxes, 'DOM' and 'Sound', both of which are unchecked.
- Code**: A code editor area with a line number '1' visible.
- Assets (zip file)**: A text input field and a 'Choose File' button.
- Desa els canvis**: A blue button at the bottom left.

Aquí hi trobem una sèrie d'opcions:

- A 'Sketch width' i 'Sketch height', hi haurem d'entrar la mida que tindrà el nostre *canvas*.
- Si necessitem incorporar *addons* de p5.js, els hem de seleccionar al camp 'Addons'.
- Al camp 'Code' disposem d'un editor de codi on haurem d'entrar el nostre programa.
- Del camp 'Assets' en parlarem a l'apartat 2.1.
- Del camp 'Container id' en parlarem a l'apartat 2.2.

Hem de recordar de fer clic a 'Desa els canvis' per guardar les modificacions que fem.



Figura 9. Exemple de codi a l'editor de 'p5.js Embed'

```

1 let y = 100;
2
3 // The statements in the setup() function
4 // execute once when the program begins
5 function setup() {
6   // createCanvas must be the first statement
7   createCanvas(720, 400);
8   stroke(255); // Set line drawing color to white
9   frameRate(30);
10 }
11 // The statements in draw() are executed until the
12 // program is stopped. Each statement is executed in
13 // sequence and after the last line is read, the first
14 // line is executed again.
15 function draw() {
16   background(0); // Set the background to black
17   y = y - 1;
18   if (y < 0) {
19     y = height;
20   }
21   line(0, y, width, y);
22 }
23

```

Mentre no necessitem integrar-lo en el WordPress, ens serà més còmode treballar el codi en un arxiu a part, com amb qualsevol altre projecte de p5.js. Quan vulguem veure el resultat en context, copiem i enganxem el programa a l'editor de WordPress. Si necessitem fer retocs o volem fer algunes proves amb la informació de WordPress que l'extensió ens proporciona (trobareu més informació sobre això a l'apartat 3), llavors potser preferim editar el codi directament en aquest editor.

Només disposem d'un camp de text per al nostre codi, així que l'haurem d'agrupar tot en un únic espai. És recomanable, doncs, separar diferents seccions amb comentaris que ens permetin distingir-les fàcilment.

## 2.1. Incorporar altres recursos

Pot ser que en el nostre programa vulguem incorporar imatges, tipografies, documents amb dades o qualsevol altre tipus de document. Per fer-ho disposem d'un camp 'Assets', on podem seleccionar un arxiu ZIP que contingui tots els recursos que volem incorporar.

Si fem clic al botó 'Choose file', se'ns obre una finestra. A la pestanya 'Penja fitxers' (figura 10) hi podem seleccionar un nou arxiu per penjar, i a la pestanya 'Mediateca' (figura 11), un arxiu que haguem pujat anteriorment. Això ens permet pujar arxius amb diferents versions dels recursos que volem fer servir i alternar-los. Si volem modificar algun dels recursos que hem pujat o afegir-ne més, però, haurem de pujar de nou un fitxer amb tots els recursos.

Figura 10. Pestanya per penjar un nou arxiu amb recursos

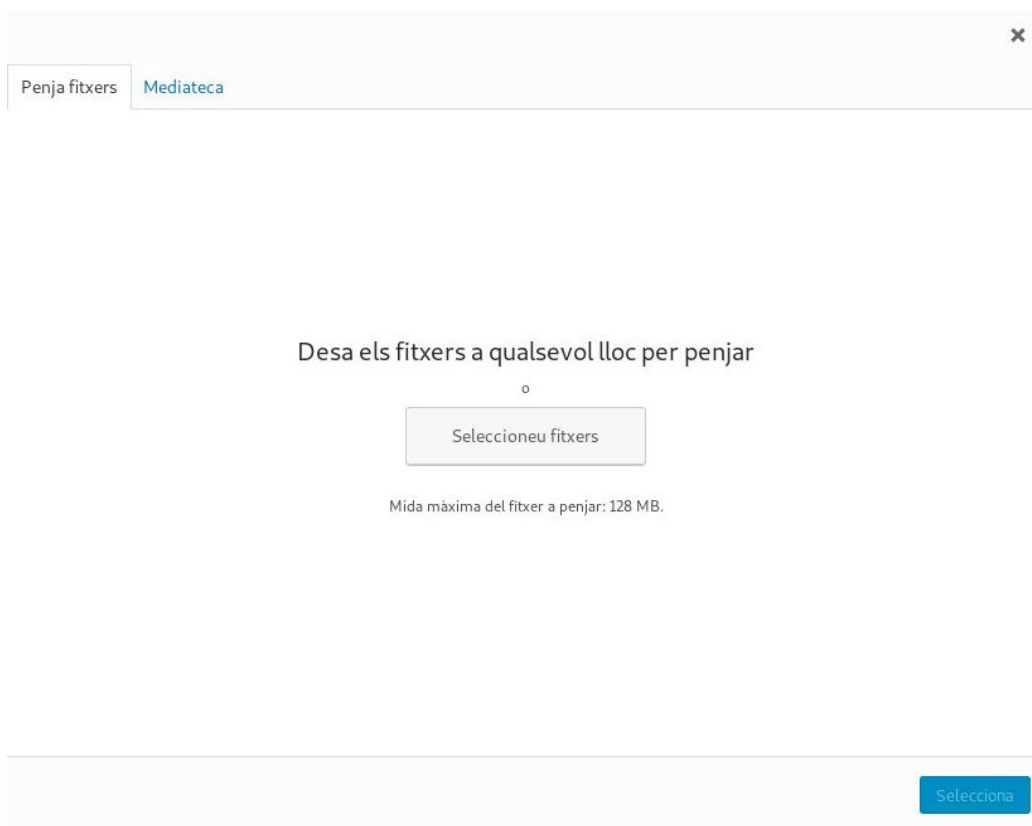
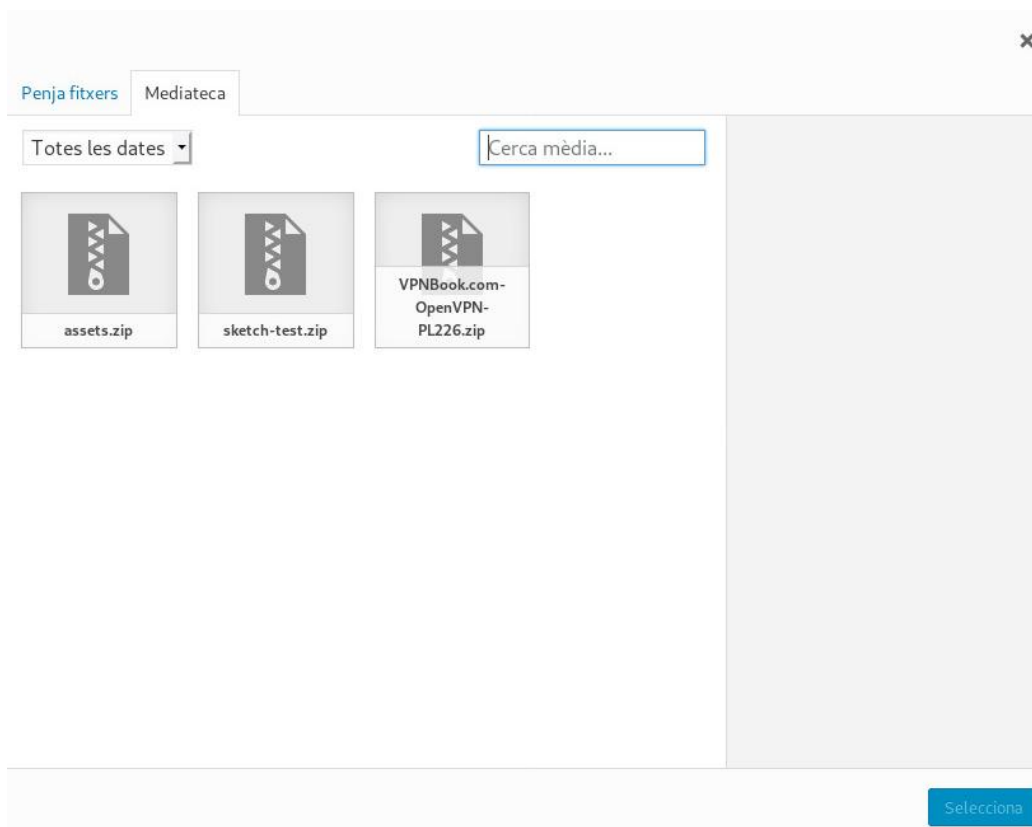


Figura 11. Pestanya per seleccionar un arxiu de recursos que haguem pujat prèviament



A l'hora de comprimir en un fitxer ZIP els arxius que volem pujar, hem de tenir en compte que hem de respectar l'estructura de carpetes que fem servir dins el codi per accedir als documents. És a dir, si tenim una imatge *fons.png* dins un directori *recursos* a la qual accedim des del nostre codi amb la ruta "*recursos/fons.png*" (ex: *var fons = loadImage("recursos/fons.png");*) haurem de comprimir la carpeta *recursos* i no només els fitxers que conté. D'aquesta manera es respectarà la mateixa ruta d'accés quan pugem el fitxer.

Igualment, si els recursos han d'estar al directori principal del nostre projecte (ex: *var fons = loadImage("fons.png");*), comprimirem directament tots els documents que vulguem pujar en un mateix arxiu ZIP.

Per comoditat, recomanem que disposeu de tots els recursos que hagueu de fer servir en el vostre programa en un directori específic i que el comprimeu quan hagueu de pujar els documents.

## 2.2. Mostrar el nostre programa

Un cop haguem introduït el codi del nostre programa, haurem de mostrar-lo en algun lloc. Tenim dues opcions per fer-ho:

1. Fer servir l'atribut *id* d'un element HTML dins el qual el volem incrustar.
2. Afegir un *shortcode* o 'Codi de substitució' en qualsevol pàgina o *post* del nostre WordPress.

### 2.2.1. Fer servir l'*id* d'un element HTML

Tot i que aquesta opció és més adient si volem integrar el programa en el disseny general del lloc web, cal estar una mica familiaritzat amb el funcionament HTML i dels temes de WordPress. Si no és el cas, és més senzill optar pel *shortcode*.

Si el nostre tema de WordPress disposa d'un element HTML dins el qual volem incrustar el nostre *sketch*, podem fer-ho si aquest element disposa d'un atribut *id* o si nosaltres mateixos som capaços d'afegir-l'hi. Llavors només cal que introduïm aquest *id* en el camp 'Container id' de les opcions de 'p5.js Embed'. D'aquesta manera el nostre *sketch* s'incrustarà al final del contingut d'aquest element. Si el tema té l'estructura adequada, això ens permet incorporar el nostre programa dins d'algun dels elements comuns del lloc web, com pot ser la capçalera. Generalment necessitarem un tema que ja estigui adaptat per a aquesta finalitat, o ser capaços nosaltres d'adaptar-lo, ja que, si no, segurament no disposarem de l'espai o la distribució necessaris per incorporar el nostre *sketch*.

Figura 12. Si la nostra pàgina disposa d'un element HTML amb un atribut *id*, hi podem incrustar el nostre sketch. En aquest cas l'*id* "logo"



Figura 13. Exemple d'element HTML amb l'*id* "logo" on podríem incrustar el nostre sketch

```

<div class="site-branding">-
  <p class="site-title"><a href="http://work.lcl/test/" rel="h
  <p class="site-description">-
    Altre lloc gestionat amb el WordPress-
  </p>-
  <div id="logo">-
    <!-- Aquest element contindrà el nostre Sketch -->-
  </div>-
  <nav id="site-navigation" class="main-navigation" aria-label
    <div class="menu-menu-container">-
      <ul id="menu-menu" class="main-menu">-
        <li id="menu-item-16" class="menu-item menu-item
          <a href="http://work.lcl/test/" aria-current
        </li>-

```

## 2.2.2. Fer servir un *shortcode*

Un *shortcode*, traduït com a 'Codi de substitució' en català, és un petit codi que ens permet incrustar determinats elements dins el contingut de les pàgines i els *posts* de WordPress.

### Nota

Els *posts* de WordPress s'anomenen 'Entrades' en català, però en aquest text farem servir el terme en anglès.

Els *shortcodes* s'escriuen entre claudàtors ([ ]). Si volem incrustar el nostre *sketch* en el contingut de qualsevol pàgina o *post*, només cal que hi incloquem el codi `[p5js_embed]`. Podem escriure'l directament al text de la pàgina (figura 14) o, si fem servir l'editor de blocs de les últimes versions de WordPress, podem afegir un bloc 'Codi de substitució' on escriurem el mateix codi (figura 15).

Figura 14. *Shortcode* escrit directament al text d'una pàgina

| LOS ANGELES, un cop un gran gos de nom Pep i m'agrada la pinya colada. (i em va a pillar la pluja.)

... o alguna cosa així:

[p5js\_embed]

| La companyia Aparells XYZ es va crear el 1971 i ha estat proporcionant aparells de qualitat al públic

Figura 15. *Shortcode* escrit dins un bloc 'Codi de substitució'

| LOS ANGELES, un cop un gran gos de nom Pep i m'agrada la pinya colada. (i em va a pillar la pluja.)

... o alguna cosa així:

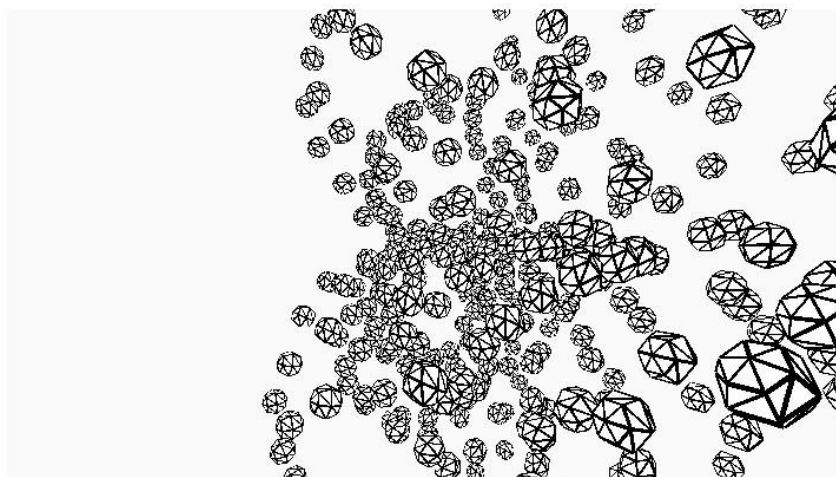
[/] Codi de substitució [p5js\_embed]

| La companyia Aparells XYZ es va crear el 1971 i ha estat proporcionant aparells de qualitat al públic

Figura 16. Resultat: *Sketch* incrustat a la pàgina, allà on hem inclòs el *shortcode*

| va a pillar la pluja.)

... o alguna cosa així:



| La companyia Aparells XYZ es va crear el 1971, i ha estat proporcionant aparells de qua-

## 3. Accedir a les dades sobre el lloc web

'p5.js Embed' ens facilita una sèrie de dades sobre el lloc web WordPress a les quals podem accedir des del nostre codi de p5.js. Per fer-ho només cal que consultem la variable `wp_data`, que és definida per l'extensió abans que s'executi el nostre programa.

`wp_data` és un objecte que consta de les propietats següents:

- **site\_title:** (String) Títol del lloc web.
- **site\_description:** (String) Descripció del lloc web.
- **post\_count:** (Number) Nombre total de *posts* publicats al lloc.
- **first\_post:** (Object) Primer *post* que es va publicar. És un objecte amb l'estructura següent:
  - **id:** (Number) Identificador numèric únic per a cada *post*.
  - **title:** (String) Títol del post.
  - **date:** (Number) Data de creació del post en milisegons des de l'època Unix (1 de gener de 1970 a les 00.00.00 h). Pensat per fer servir amb la classe `Date` de JavaScript.
  - **type:** (String) Tipus de *post*. En aquest cas serà sempre "post".
  - **author:** (Object) Objecte amb la informació següent sobre l'autor del *post*:
    - **id:** (Number) Identificador numèric únic per a cada autor.
    - **name:** (String) Nom de l'autor.
    - **count:** (Number) Nombre de *posts* publicats per aquest autor.
  - **categories:** (Array) *Array* de les categories assignades al *post*. Cada element de l'*array* té l'estructura següent:
    - **id:** (Number) Identificador numèric únic per a cada categoria.
    - **name:** (String) Nom de la categoria.
    - **count:** (Number) Nombre de *posts* als quals s'ha assignat aquesta categoria.
    - **type:** (String) Tipus de taxonomia. En aquest cas serà sempre "category".

- **parent:** (Number) Identificador numèric de la ‘categoria mare’ d’aquesta. Les categories són jeràrquiques i, per tant, pot ser que una categoria sigui subcategoria d’una altra. En cas que es tracti d’una categoria de primer nivell, aquest valor serà 0.
- **tags:** (Array) Array dels *tags* (‘etiquetes’ en català) assignats al *post*. Té la mateixa estructura que ‘categories’, però en aquest cas ‘type’ és sempre “post\_tag” i ‘parent’ és sempre 0.
- **last\_post:** (Object) Últim post que s’ha publicat. Té la mateixa estructura que ‘first\_post’.
- **categories:** (Array) Llista de totes les categories que hi ha. Té la mateixa estructura que la propietat ‘categories’ de ‘first\_post’ i ‘last\_post’.
- **tags:** (Array) Llista de tots els *tags* que hi ha. Té la mateixa estructura que la propietat ‘tags’ de ‘first\_post’ i ‘last\_post’.
- **current:** (Object) Conté informació sobre la pàgina actual. Les seves propietats poden variar segons el tipus de contingut que estiguem visualitzant. Aquestes són les possibilitats:
  - **type:** (String) Ens indica de quin tipus de contingut es tracta. Les opcions són:
    - “**posts\_page**”: És la pàgina amb la **llista de posts** publicats.
    - “**post**”: És la pàgina específica d’un **únic post**.
    - “**page**”: És una **pàgina genèrica**.
    - “**category**”: És la llista de *posts* per a una **categoria** concreta.
    - “**post\_tag**”: És la llista de *posts* per a un **tag** concret.
    - “**author**”: És la llista de *posts* d’un **autor**.
    - “**date**”: És la llista de *posts* d’una **data o rang de dates** concretes.
    - “**404**”: Significa que l’adreça a la qual s’ha accedit no s’ha trobat, perquè no es correspon amb cap pàgina del lloc.
  - **page:** (Number) Ens indica a quin número de pàgina del contingut que estem visualitzant ens trobem, ja que pot ser que les llistes de *posts* o els continguts de les pàgines s’extenguin al llarg de diversos números de pàgina. Comença a comptar des d’1, però si marca 0 vol dir que no s’ha especificat cap número de pàgina, i per tant, equival a 1.
  - **page\_title:** (String) Títol de la pàgina tal com apareix a la pestanya o a la finestra del navegador. Només la part que fa referència al títol de la pàgina actual.



- **tab\_title:** (String) Títol complet que apareix a la pestanya o a la finestra del navegador. Generalment té un format tipus “Títol de la pàgina - Títol del lloc”, mentre que a la pàgina principal del lloc acostuma a ser tipus “Títol del lloc - Descripció del lloc”.
- **is\_home:** (Boolean) Ens indica si la pàgina actual és la pàgina principal del lloc (la portada) o no.
- **posts:** (Array) Només apareix si la pàgina actual és una llista de *posts*. Conté un ‘Array’ amb tots els *posts* de la llista. Cadascun dels elements té la mateixa estructura que ‘first\_post’ o ‘last\_post’.

A part de les propietats ja descrites, quan ‘current’ sigui de tipus “**post**” o “**page**” tindrà les mateixes propietats que ‘first\_post’ o ‘last\_post’, referents al *post* o la pàgina actual.

A més, si és de tipus “**page**” tindrà també una propietat ‘**parent**’ amb l’identificador numèric de la pàgina superior (o 0 si no n’hi ha), ja que les pàgines poden ser jeràrquiques.

Si ‘current’ és de tipus “**category**” o “**post\_tag**”, tindrà les mateixes propietats que els elements dels ‘Arrays’ ‘categories’ o ‘tags’.

Si és del tipus “**author**”, tindrà les mateixes propietats que la propietat ‘author’ de ‘first\_post’ o ‘last\_post’.

Si ‘current’ és de tipus “**date**”, tindrà una propietat ‘**vars**’, que és un objecte les propietats del qual poden ser una o més de les següents: ‘**year**’, ‘**month**’, ‘**day**’, ‘**hour**’, ‘**minute**’ i ‘**second**’. La data de publicació dels *posts* de la llista coincidirà amb els valors d’aquestes variables temporals (les que hi siguin presents). Per exemple, si estem consultant una llista de *posts* publicats l’abril de 2015, ‘vars’ tindrà el format següent: { year: 2015, month: 4 }.

Així doncs, si volem obtenir el nombre total de *posts* publicats consultarem **wp\_data.post\_count**. Si sabem que som en una pàgina que és una llista de *posts* (perquè, per exemple, “**date**” === **wp\_data.current.type**) i volem saber quants se’n mostren a la pàgina actual, podem consultar **wp\_data.current.posts.length**. O si volem saber quants *posts* ha publicat l’autor del darrer *post*, ho trobarem a **wp\_data.last\_post.author.count**.

Us recomanem que, si no us queda clara l’estructura d’aquestes dades, o si voleu consultar-ne tota la informació per a una pàgina concreta, feu **console.log( wp\_vars )** al principi del vostre codi per mostrar-ne el contingut a la consola del navegador.