

**Esther Navarro Bravo**

**Visualització de dades del projecte ACTUA amb un CMS**

**TREBALL DE FI DE GRAU**

**dirigit pel Dr. Antoni Martínez Ballesté**

**Grau d'Enginyeria de Sistemes i Serveis de Telecomunicacions**



**UNIVERSITAT ROVIRA I VIRGILI**

**Tarragona**

**2022-2023**

## **Resum**

Aquest treball fi de grau té per objectiu principal analitzar el concepte de CMS (Content Management System). En concret, s'estudien els CMS Oracle WebCenter Sites i Wordpress, cadascun amb àmbits d'aplicació diferents. Addicionalment s'aborda el disseny i desenvolupament d'un connector que permeti obtenir i visualitzar dades del servidor del projecte ACTUA, fent consultes a la seva API. Aquest projecte ha desplegat vora un centenar de kits de monitorització del context en aules de tota Catalunya. A banda de contenir una anàlisi de les dues eines esmentades, el treball inclou el disseny, implementació i prova d'aquest connector.

## **Resumen**

Este trabajo fin de grado tiene por objetivo principal analizar el concepto de CMS (Content Management System). En concreto, se estudian los CMS Oracle WebCenter Sites y Wordpress, cada uno con distintos ámbitos de aplicación. Adicionalmente se aborda el diseño y desarrollo de un conector que permita obtener y visualizar datos del servidor del proyecto ACTUA, realizando consultas a su API. Este proyecto ha desplegado cerca de un centenar de kits de monitorización del contexto en aulas de toda Cataluña. Aparte de contener un análisis de las dos herramientas mencionadas, el trabajo incluye el diseño, implementación y prueba de este conector.

## **Abstract**

The main objective of this thesis is to analyze the concept of CMS (Content Management System). Specifically, the CMS Oracle WebCenter Sites and Wordpress are studied, each with different fields of application. In addition, the design and development of a plugin that allows data to be obtained and visualized from the ACTUA project server, by querying its API, is addressed. This project has deployed close to a hundred context monitoring kits in classrooms throughout Catalonia. Apart from containing an analysis of the two tools mentioned, the work includes the design, implementation and testing of this connector.

## Taula de continguts

<b>1. Introducció</b> .....	<b>5</b>
<b>2. Fonaments</b> .....	<b>6</b>
<b>2.1. HTML</b> .....	<b>6</b>
2.1.1. Evolució de l'HTML .....	6
2.1.2. Fonaments d'HTML .....	7
2.1.3. Estructura .....	8
<b>2.2. Java</b> .....	<b>9</b>
2.2.1. Evolució de Java.....	9
2.2.2. Característiques principals .....	9
2.2.3. Normes a seguir.....	10
2.2.4. Separadors.....	10
<b>2.3. JSP</b> .....	<b>10</b>
2.3.1. Història del JSP .....	11
2.3.2. Avantatges del JSP .....	11
2.3.3. Estructura del JSP .....	12
<b>3. Part teòrica</b> .....	<b>13</b>
<b>3.1. CMS</b> .....	<b>13</b>
3.1.1. Beneficis principals d'un CMS.....	13
3.1.2. Funcionament d'un CMS .....	14
3.1.3. CMS utilitzats al mercat.....	14
<b>3.2. Oracle WebCenter Sites</b> .....	<b>15</b>
3.2.1. Evolució de Oracle WebCenter Sites .....	15
3.2.2. Fonaments bàscis .....	15
3.2.3. Funcionament.....	16
3.2.4. Avantatges i inconvenients.....	17
<b>3.3. WordPress</b> .....	<b>18</b>
3.3.1. Evolució de Wordpress .....	19
3.3.2. Estructura .....	19
3.3.3. Avantatges i inconvenients.....	20
<b>3.4. Elecció d'un CMS</b> .....	<b>21</b>
<b>4. Part pràctica</b> .....	<b>23</b>
<b>4.1. Cas d'ús</b> .....	<b>23</b>
4.1.3. Informació sobre l'API .....	23
<b>4.2. Oracle WebCenter Sites</b> .....	<b>24</b>

4.2.1.	Desenvolupament tècnic.....	24
4.2.2.	Gestió dels continguts .....	41
4.2.3.	Resultat final del lloc web.....	50
<b>4.3.</b>	<b>Wordpress.....</b>	<b>63</b>
4.3.1.	Instal·lació.....	63
4.3.2.	Funcionament.....	64
4.3.3.	Resultat final del lloc web.....	64
<b>5.</b>	<b>Avaluació .....</b>	<b>70</b>
<b>5.1.</b>	<b>Avaluació Oracle WebCenter Sites .....</b>	<b>70</b>
<b>5.2.</b>	<b>Avaluació Wordpress.....</b>	<b>70</b>
<b>6.</b>	<b>Conclusió.....</b>	<b>72</b>
<b>7.</b>	<b>Bibliografia .....</b>	<b>74</b>

## 1. Introducció

L'objectiu principal d'aquest estudi és analitzar quina és la millor tecnologia per tal de la creació d'un lloc web. Per fer-ho, estudiarem dues tecnologies molt potents actualment al mercat, Oracle WebCenter Sites i Wordpress. Per començar, examinarem cada una d'elles d'una forma teòrica i seguidament, realitzarem la mateixa web amb cadascuna de les tecnologies esmentades. D'aquesta manera podrem concloure el nostre estudi, sabent els avantatges i inconvenients de cadascuna d'elles. Aprofitarem la creació del lloc web per mostrar els resultats d'un projecte dut a terme per un grup d'investigadors de la nostra universitat, el projecte Actua que consisteix en analitzar dades per investigar la transmissió del SARS-CoV-2, el virus causant de la COVID-19, a les aules de les escoles.

Pel que fa a les especificacions, es plantegen les següents:

- Entendre què és un sistema de gestió de continguts (CMS)
- Analitzar el funcionament dels CMS
- Valorar concretament els CMS d'*Oracle Web Center Sites* i *Wordpress*
- Analitzar profundament dos sistemes de gestió de continguts
- Comprendre les característiques, avantatges i àmbit d'aplicació de cadascun dels CMS analitzats
- Realització d'una web centrant-nos en la part *back-end* en dos sistemes diferents
- Comparació entre dos sistemes de continguts
- Mostrar els resultats del projecte Actua

La realització de les pràctiques acadèmiques a una multinacional del sector de les tecnologies de la informació i la comunicació, m'ha permès desenvolupar i implementar solucions tecnològiques, concretament en lloc webs. Fins al moment de començar tenia absolut desconeixement per aquest gran tema i és aquest el principal motiu que m'ha portat a la realització d'aquest treball fi de grau.

## 2. Fonaments

Per la realització pràctica d'aquest estudi, hem de tindre clar diferents aspectes, així com el principal llenguatge de marcatge, HTML, el principal llenguatge de programació, Java i per últim, la tecnologia Java Server Pages, orientada a la creació de pàgines web.

### 2.1. HTML

HTML és un llenguatge de Marques d'Hipertext, de l'anglès, *HyperText Markup Language*, cal recalcar que no és un llenguatge de programació, sinó un llenguatge de marcatge que defineix l'estructura d'un contingut.

Podem afirmar que l'HTML és el component més bàsic que forma una pàgina Web. Es compon a base d'etiquetes, també anomenades marques o tags, amb les quals aconseguim expressar cadascuna de les parts d'un document, com pot ser la capçalera, el cos principal, els seus paràgrafs, etc.

L'HTML és el llenguatge principal que hem de posar en pràctica a l'hora de construir un lloc web, sigui quina sigui la tecnologia, eina o gestor de contingut que utilitzem sempre utilitzarem HTML com a estructura bàsica i a partir d'aquí, ja podrem afegir-hi unes altres funcionalitats.

#### 2.1.1. Evolució de l'HTML

L'origen de l'HTML el traslladem al 1980 quan el físic Tim Berners-Lee va proposar un nou sistema d'hipertext per a compartir documents. No va ser fins al 1991 quan es va publicar el primer document formal amb la descripció d'HTML, aquest document es va anomenar *HTML Tags* i encara avui en dia es pot consultar de manera digital a mode de relíquia informàtica. En primer moment, es van descriure una petita quantitat de tags o etiquetes que tenien la utilitat de compondre documents, però la innovació més grossa va ser la possibilitat de crear enllaços entre documents i d'aquesta manera fer possible la navegació entre diferents pàgines.

Per a determinar el llenguatge HTML es va utilitzar com a base SGML, de l'anglès *Standard Generalized Markup Language*, que va permetre especificar uns altres llenguatges de marcatge. No va ser fins al 1993 quan es va publicar el primer esbós d'especificació del llenguatge HTML basat en SGML i va ser creada en el marc de la IEFT, en anglès Internet Engineering Task Force. Tot aquest procés és conegut com a HTML 2.0

A continuació, el 1996 es va crear l'organització que actualment és l'encarregada de regular els estàndards de la web, coneguda com a W3C<sup>1</sup>. Després d'aquest fet, l'any 1997 es va publicar la versió 3.2 d'HTML, que va ser la primera recomanació d'HTML publicada pel W3C.

Seguidament, es va publicar la versió 4.0 al 1998 que va suposar un gran salt de les versions anteriors. Els canvis més significatius es troben a les fulles d'estils en cascada CSS<sup>2</sup>, i la possibilitat d'incloure petits programes a les pàgines webs. Posteriorment, es va crear el WHATWG, de l'anglès *Web Hypertext Application Technology Working Group*, un grup paral·lel centrat en desenvolupar les tecnologies de la web. Gràcies a aquest grup, l'HTML va tindre un gran impuls i finalment, es va

---

<sup>1</sup> World Wide Web Consortium

<sup>2</sup> Cascading Style Sheets

arribar a publicar l'HTML 5, creant els primers documents d'estandardització de HTML 5 en 2007.

### 2.1.2. Fonaments d'HTML

Com ja s'ha exposat anteriorment, quan escrivim codi HTML estem escrivint etiquetes HTML. Tots aquests elements tenen unes parts específiques que s'han de complir:

- L'etiqueta de obertura: consisteix en el nom del element tancat per parèntesis angulars (< >) de obertura i de tancament. Estableix on comença i on acaba a tindre efecte l'element.
- L'etiqueta de tancament: es igual a l'etiqueta de obertura però inclou una barra de tancament (/) abans del nom de l'etiqueta. Estableix on acaba l'element.
- El contingut: es el contingut del element.

Els elements també poden tindre atributs, aquest són els que contenen la informació addicional del element, aquesta no volem que aparegui en el contingut real del element. Els atribut els inclouem sempre en l'etiqueta d'apertura de l'element. Un atribut ha de tenir sempre:

- Un espai entre aquest i el nom de l'element
- El nom de l'atribut seguit d'un signe igual (=)
- Cometes d'apertura i de tancament.

A continuació s'exposaran algunes de les etiquetes més utilitzades per a crear qualsevol document HTML:

- `<body></body>`

Indica la part del cos del contingut de un document HTML. Aquesta etiqueta és essencial per a qualsevol document ja que indica on comença el contingut visible del document.

- `<head></head>`

A la part superior del document HTML es on indicarem les metadades, com per exemple el títol del document, fulles d'estil, CSS, etc.

- `<div></div>`

Aquest element és utilitzat normalment per agrupar uns altres elements, és a dir, ens ajuda a estructurar un document en seccions i que sigui més entenedor.

- `<a></a>`

Es una etiqueta que ens ajuda a crear un enllaç a una altra pàgina web. Sempre va acompanyada de l'atribut *href* on introduïrem l'enllaç al qual volem connectar.

- `<strong></strong>`

Aquesta etiqueta l'hi dona èmfasi a una part en concret del text en el cas que vulguem remarcar-la.

- `<br>`

Gràcies a aquesta etiqueta podem introduir un salt de línia i que el text sigui més llegible.

- `<h1></h1>...<h6></h6>`

Hi ha diferents nivells de títols, des del 1 fins al 6. Aquestes etiquetes ens permeten indicar la rellevància del títol de major a menor. També ajudem als *bots* a entendre la importància del contingut.

- `<img></img>`

Utilitzem aquesta etiqueta per a visualitzar imatges dins d'un contingut. Hem d'incloure obligatòriament l'atribut *src*, on indicarem la ruta de la imatge.

- `<ol> <li></li></ol> | <ul> <li></li></ul>`

Aquestes etiquetes ens serveixen per la creació de llistes, `<ol>` pel cas de llistes ordenades i `<ul>` per desordenades. Cadascun dels continguts de la llista s'identifiquen amb l'etiqueta `<li>`.

- `<p></p>`

L'etiqueta `<p>` ens serveix per agrupar text dins d'un mateix paràgraf.

### 2.1.3. Estructura

Una vegada tenim clars una sèrie de conceptes bàsics, hem de saber quina és la estructura que sempre haurem de seguir i, a partir d'aquí, desenvolupar.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mi pagina de prueba</title>
  </head>
  <body>
    
  </body>
</html>
```

Cal comentar algunes de les parts de l'estructura:

- `<!DOCTYPE html>`

Indica el tipus de document.

- `<html></html>`

Aquest element és l'encarregat de contenir tot el contingut de la pàgina web sencera.

- `<meta charset="utf-8">`

Aquest element ens indica que tots els caràcters del document utilitzaran UTF-8.

- `<title></title>`

Estableix el títol de la pagina i aquest apareix en la pestanya del navegador quan es carrega esmentada pàgina.

## 2.2. Java

Java és el llenguatge de programació i desenvolupament número u d'avui en dia. Es caracteritza per la reducció de costos, escurça els terminis de desenvolupament, impulsa la innovació i millora els serveis de les aplicacions.

### 2.2.1. Evolució de Java

Java es va crear com una eina de programació per utilitzar-la només en un projecte de televisió. El seu nom original va ser *Oak*, després va rebre el nom de *Green* fins que finalment la van anomenar com la coneixem avui en dia, Java. Actualment, no es coneix el significat real del seu nom, però es tenen diverses hipòtesis, entre elles les inicials d'alguns dels seus dissenyadors, l'acrònim de *Just Another Vague Acronym*, el nom d'un tipus de cafè d'una cafeteria on solien anar els seus creadors o el van extreure d'una llista aleatòria de paraules.

Els criteris que van imposar des d'un principi per dissenyar Java van ser:

- Independent de la màquina
- Sense risc per treballar en xarxa
- Potent per a que es pugui substituir per codi natiu

Des del seu inici hi ha hagut diverses versions, sent la primera *JDK 1.0* llançada al mercat el gener del 1996, fins arribar a l'actual, Java SE 18, disponible des del març de 2022.

### 2.2.2. Característiques principals

Les principals característiques de Java són les següents:

- És un llenguatge compilat i interpretat, això vol dir que tot programa en Java s'ha de compilar i el codi resultant és interpretat per una màquina virtual.
- Està orientat a objectes, és a dir, els diferents tipus de dades que s'utilitzen estan lligades a les operacions. D'aquesta manera, les dades i el codi es combinen en diferents objectes. L'objectiu principal és que grans projectes siguin fàcils de gestionar.
- Gran seguretat. Java l'hi dona molta importància a la seguretat i existeixen diferents nivells de seguretat, des del programador fins a la interpretació de la màquina virtual.
- Existeix un recollidor de brossa, aquest és l'encarregat d'alliberar la memòria ocupada per objectes que ja no estan referenciats.
- El programador sempre està obligat a tractar les excepcions que es puguin produir durant l'execució del codi; existeixen diferents procediments predefinits per Java per tractar aquests errors.
- Existeixen una gran quantitat de classes i aquestes tenen el seu propi espai reservat de noms per tal que no interfereixin amb les noves classes creades per l'usuari.
- Java és un dels llenguatges amb major creixement i amplitud d'utilització en diferents àmbits dins de la tecnologia. Alguns d'aquests àmbits són:
  - Dispositius mòbils i sistemes embastats
  - Navegador web
  - Sistemes de servidor

- Aplicacions d'escriptori

### 2.2.3. Normes a seguir

Per poder utilitzar correctament el llenguatge Java és necessari el coneixement de diferents normes de codificació, d'aquesta manera serem capaços de llegir i entendre qualsevol codi d'un altre programador. Aquestes regles són les següents:

- Els noms de les classes han de començar per majúscula.
- Les constants han d'estar escrites en majúscules.
- Els atributs i mètodes de les classes han de començar per minúscules i en format *CamelCase*.

### 2.2.4. Separadors

En el llenguatge Java tenim sis separadors diferents i cadascun d'ells tenen unes funcions específiques.

1. Els parèntesis “( )”
  - Ubicació dels paràmetres d'un mètode
  - Modifiquen la procedència d'una expressió
  - Delimiten les condicions
2. Els claudàtors “[ ]”
  - Declaren vectors i permeten accedir als seus elements
3. Les claus “{ }”
  - Defineixen blocs de codi
  - Delimiten la llista de valors inicials d'una cadena de valors
4. El punt “.”
  - Accedeixen als atributs i mètodes d'una classe
5. La coma “,”
  - Separen paràmetres dins d'un mètode
  - Separen variables en la seva declaració
6. El punt i la coma “;”
  - Finalitzen instruccions

## 2.3. JSP

La tecnologia Java Server Pages, més coneguda com a JSP, a grans trets, està orientada a crear pàgines web amb programació en Java.

Amb JSP els dissenyadors de webs poden incorporar elements dinàmics dins de la pàgina utilitzant porcions de codi Java incrustat o diferents etiquetes. Per tant, les pàgines JSP tenen un aspecte molt semblant al d'una pàgina d'HTML, on a aquest se li ha introduït codi Java amb les seves corresponents etiquetes. D'aquesta manera, quan una pàgina és sol·licitada per un usuari i processada per un servidor HTTP, el codi HTML serà enviat directament a l'usuari, però el codi Java serà executat pel servidor quan es rebi la sol·licitud, d'aquesta manera es generarà el contingut dinàmic de la pàgina web. Quan l'usuari vulgui visualitzar el codi font de la pàgina, només veura el HTML ja que el codi JSP ja haurà sigut executat.

### **2.3.1. Història del JSP**

JSP (JavaServer Pages) és una tecnologia de programació web que s'utilitza per a crear pàgines web dinàmiques en Java. Va ser creada per Sun Microsystems en 1999 i després va ser adoptada per Oracle Corporation quan va adquirir Sun Microsystems en 2010.

Abans de JSP, la majoria dels desenvolupadors web feien servir tecnologies com CGI (Common Gateway Interface) i ASP (Active Server Pages) per a crear aplicacions web dinàmiques. No obstant això, aquestes tecnologies eren limitades en termes d'escalabilitat i manteniment.

JSP es va crear per a superar aquestes limitacions, oferint als desenvolupadors web una forma més eficient i escalable de crear aplicacions web dinàmiques. La tecnologia JSP permet als desenvolupadors web combinar HTML, Java i altres tecnologies web com JavaScript i CSS per a crear pàgines web dinàmiques.

El procés de desenvolupament d'una pàgina web amb JSP implica la creació d'arxius JSP, que contenen codi Java incrustat en una pàgina HTML. Quan un usuari sol·licita una pàgina web, el servidor JSP processa la sol·licitud i genera una pàgina web dinàmica, que es retorna a l'usuari.

Des de la seva creació, JSP ha evolucionat significativament i ha estat àmpliament adoptat en la indústria de desenvolupament web. S'ha convertit en una tecnologia essencial per a la creació d'aplicacions web a Java, i continua sent una eina valuosa per als desenvolupadors web que busquen crear aplicacions web escalables i de senzill manteniment.

### **2.3.2. Avantatges del JSP**

Els avantatges d'utilitzar JSP (JavaServer Pages) en el desenvolupament d'aplicacions web són diverses. A continuació, es presenten alguns dels principals avantatges de JSP:

- Fàcil d'aprendre: JSP és una tecnologia fàcil d'aprendre per als desenvolupadors web que ja tenen coneixements de Java i HTML.
- Separació de la lògica de negoci i la presentació: JSP permet separar la lògica de negoci de la presentació, la qual cosa facilita la modificació i el manteniment del codi.
- Reutilització de codi: JSP permet reutilitzar el codi Java en diverses pàgines web, la qual cosa facilita el desenvolupament i redueix el temps de codificació.
- Integració amb altres tecnologies web: JSP s'integra fàcilment amb altres tecnologies web, com JavaBeans, Servlets, JDBC i altres.
- Escalabilitat: JSP és escalable i pot manejar grans volums de trànsit i sol·licituds.
- Facilita el desenvolupament d'aplicacions web dinàmiques: JSP permet als desenvolupadors web crear aplicacions web dinàmiques que poden canviar en funció de les necessitats de l'usuari.
- Àmplia comunitat de desenvolupadors: JSP té una àmplia comunitat de desenvolupadors que comparteixen els seus coneixements i recursos en línia.

En resum, JSP és una tecnologia de programació web escalable, fàcil d'aprendre i que permet als desenvolupadors web crear aplicacions web dinàmiques i reutilitzar el codi. Tot això ho converteix en una eina valuosa per al desenvolupament d'aplicacions web empresarials.

### 2.3.3. Estructura del JSP

Un arxiu JSP (JavaServer Pages) es compon de diverses seccions, algunes de les quals són opcionals. L'estructura bàsica d'un arxiu JSP es pot dividir en tres seccions principals:

- Declaracions: S'utilitza per a declarar variables, mètodes i classes que s'utilitzaran en la resta del JSP. Les declaracions s'escriuen dins de les etiquetes `<%! %>`.
- Directives: Són instruccions especials que li indiquen al servidor JSP com processar l'arxiu JSP. Les directives comencen amb l'etiqueta `<%@` i acaben amb `%>`. Hi ha diverses directives disponibles en JSP, però les més comunes són la directiva `page` i la directiva `include`.
- Contingut: És la secció principal del JSP que es mostra en el navegador web. El contingut s'escriu dins de l'etiqueta `<% %>`. Pot contenir tant codi Java com HTML. En la majoria dels casos, la major part del contingut s'escriurà en HTML perquè el navegador pugui mostrar la pàgina correctament.

A continuació es mostra un exemple bàsic de l'estructura d'un arxiu JSP:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html>
<html>
  <head>
    <title>Exemple de JSP</title>
  </head>
  <body>
    <%
      // Codi Java aquí
      String nom = "Esther";
      out.println("Hola " + Esther);
    %>
  </body>
</html>
```

En aquest exemple, la directiva `page` s'utilitza per a especificar el llenguatge de programació i el tipus de contingut que s'enviarà al navegador. La secció del contingut conté codi Java que imprimeix "Hola Juan" en el navegador.

### 3. Part teòrica

En aquest apartat es descriuen els coneixements que he cercat com a resultat d'un dels objectius d'aquest treball.

#### 3.1. CMS

CMS són les sigles de *Content Management System*, en català Sistema de Gestió de Continguts. Breument, podem dir que un CMS ens permet crear, organitzar, publicar i eliminar els continguts del teu lloc web.

Un Sistema de Gestió de Continguts és un programari que presenta un entorn de treball per a la creació i administració de continguts per usuaris administradors i editors en un entorn web.

Qualsevol CMS ha de permetre les següents funcions mínimes:

- Creació, manteniment i eliminació de la informació
- Presentació i publicació de la informació
- Administració de l'entorn web

##### 3.1.1. Beneficis principals d'un CMS

Un Sistema de Gestió de Continguts té molts beneficis a l'hora de crear pàgines web. A continuació exposarem els més destacats:

###### - **Velocitat de càrrega**

La velocitat de càrrega és un dels beneficis més destacats de la utilització d'un CMS. Aquest contribueix a augmentar la velocitat de càrrega d'una pàgina web.

Podem dir que, és un element decisiu per tal de poder oferir una bona experiència al consumidor. Per tant, hem de monitorar el temps de càrrega de les webs, buscant els temps més òptim possible.

###### - **Baixos costos d'actualització i manteniment**

Sense la utilització d'un CMS, un programador tècnic pot construir un lloc web, però cada vegada que s'hagi de fer una actualització o simplement un manteniment, aquest ho haurà de fer una persona altament qualificada en les eines específiques utilitzades. És per aquest motiu que l'actualització i el manteniment d'una pàgina web sense fer ús d'un CMS resulta molt més costos.

La majoria dels softwares CMS són gratuïts, creats de forma col·laborativa arreu del món, per tant, el treball serà molt més senzill i econòmic. Això converteix CMS en una opció molt més econòmica.

###### - **Abundants recursos addicionals**

Hi ha una gran quantitat de softwares CMS, però depenent del nostre objectiu hem de saber triar correctament quin és el que millor s'adapta a les nostres necessitats.

Tots els softwares tenen una gran quantitat de recursos addicionals que podem utilitzar, aquests es poden afegir al nostre lloc web en uns senzills passes a seguir i, crear la millor experiència possible per al consumidor; de la mateixa manera, podem fer que l'administració sigui molt més eficient.

###### - **Facilitat per a crear i gestionar el seu manteniment**

Un dels grans beneficis de la utilització d'un CMS és la facilitat de creació i manteniment pel desenvolupador, ja que no fa falta descarregar cap mena de sistema concret i el treball es pot fer directament des del navegador. Això ens permet la ràpida creació de pàgines per poder oferir continguts de millor qualitat.

#### - **Fàcil utilització**

Com sabem, totes les pàgines web han de ser flexibles per a poder adaptar-nos amb facilitat a les novetats i tendències que apareixen al mercat. Gràcies a la utilització d'un CMS tindrem la capacitat per administrar els diferents tipus de continguts com poden ser imatges, vídeos, opinions dels clients, etc. Una vegada hem inclòs aquests nous continguts també ho podrem provar i optimitzar per aconseguir els millors resultats possibles amb més rapidesa.

#### - **Adaptabilitat en projectes de qualsevol classe**

Per últim, però no menys important, destaquem el benefici d'adaptabilitat en qualsevol tipus de projectes. Un CMS assoleix donar-li suport a diferents projectes de diferents mides amb la mateixa estructura i funcionament intern.

Per exemple, podem crear una pàgina per publicar diferents continguts diàriament, una web amb funcions *ecommerce*, una xarxa social o un lloc interactiu entre els diferents usuaris participants, entre d'altres.

### **3.1.2. *Funcionament d'un CMS***

El funcionament d'un CMS és molt senzill i està format per dues parts ben diferenciades: *front-end* i *back-end*. La part de *front-end* és la visible per l'usuari, és a dir, el que veiem; d'altra banda, el *back-end* és la part d'administració on es gestiona tots els continguts de la nostra pàgina web.

Tots els softwares utilitzats en gestors de continguts estan creats amb llenguatges de programació compatibles amb els navegadors webs i motors de recerca. També tenen integrats editors de text on podem crear i gestionar el contingut sense tindre coneixements del llenguatge de programació emprat. Una vegada el programador tècnic tingui una pàgina web desenvolupada, una persona sense coneixements tècnics podrà modificar quantes vegades vulgui el contingut de la pàgina en qüestió.

Podem dir que els CMS ens permeten centrar-nos més en l'estratègia i menys en la part operativa.

### **3.1.3. *CMS utilitzats al mercat***

Avui en dia hi ha una gran quantitat de sistemes de gestió de continguts en el mercat i, depenent del nostre objectiu utilitzarem un o un altre. A continuació analitzarem dos d'ells: Oracle WebCenter Sites i Wordpress, tots dos els farem servir també en la part pràctica per descobrir realment els avantatges i inconvenients que tenen.

## 3.2. Oracle WebCenter Sites

Oracle WebCenter Sites és l'eina per a gestionar una experiència Web, un sistema gestor de continguts CMS (Content Management System) el qual permet crear una estructura de suport per a la creació i administració de continguts. Permet desenvolupar llocs Webs mitjançant frameworks MVC (model vista controlador) amb l'objectiu de separar la lògica de negoci de la capa de presentació Web, és a dir, els desenvolupadors Java poden escriure la lògica i crear diferents plantilles mentre que els desenvolupadors webs poden ocupar-se de la capa de presentació escrivint HTML i JavaScript. Amb això aconseguim que no s'interfereixi el treball dels uns amb els altres.

### 3.2.1. Evolució de Oracle WebCenter Sites

Oracle WebCenter Sites és un sistema de gestió de contingut empresarial (ECM) que ha experimentat una evolució significativa des del seu inici. La història d'Oracle WebCenter Sites es remunta a mitjan dècada de 1990, quan l'empresa Vignette Corporation va llançar el seu producte Vignette StoryServer. Aquest programari revolucionari permetia a les organitzacions crear, administrar i lliurar contingut en línia de manera efectiva.

En 2009, Oracle va adquirir Vignette Corporation i, com a resultat, va obtenir el producte Vignette StoryServer. Posteriorment, Oracle ho va rebatejar com Oracle WebCenter Sites i ho va incloure com a part de la seva cartera de productes de WebCenter.

La versió inicial proporcionava una plataforma robusta per a administrar el contingut web i oferir experiències personalitzades als usuaris. Permetia la creació i gestió de llocs web dinàmics, amb característiques com la segmentació d'audiència, la gestió de fluxos de treball i la integració amb altres sistemes empresarials.

Al llarg dels anys, Oracle ha continuat millorant i ampliant les capacitats de WebCenter Sites. En versions posteriors, es van introduir funcionalitats addicionals, com la gestió d'actius digitals (DAM), la gestió de campanyes de màrqueting i la integració amb xarxes socials. Aquestes addicions van permetre a les organitzacions aprofitar encara més la plataforma per a oferir experiències digitals riques i personalitzades.

Amb el temps, Oracle WebCenter Sites també s'ha adaptat a les demandes canviants del panorama tecnològic. S'ha enfocat en el lliurament de contingut a través de diversos canals, com a dispositius mòbils i aplicacions. Això ha permès a les organitzacions estendre el seu abast i brindar experiències consistents en diferents plataformes.

A més, Oracle ha posat un fort èmfasi en la integració de WebCenter Sites amb altres productes de la seva cartera, com Oracle WebCenter Content i Oracle WebCenter Portal. Això ha permès a les organitzacions aprofitar tot el potencial de la seva infraestructura de contingut i crear solucions més completes i connectades.

### 3.2.2. Fonaments bàsics

El model de dades en WebCenter Sites es basa en *Assets*. Un asset és un objecte que és emmagatzemat en una base de dades de WebCenter. Aquest pot ser creat, modificat, inspeccionat, eliminat o duplicat segons les necessitats, podem dir que

equival a les instàncies d'un objecte. A continuació, es descriuen les característiques principals d'un Asset:

- És l'estructura de dades fonamental en WebCenter Sites.
- Els assets emmagatzemen continguts o informació estructural.
- Cada asset pertany a un *Asset Type*.
- Els assets estan formats per atributs, aquests poden ser de múltiples tipus (String, int, float, etc).
- Als assets se li poden associar altres assets flexibles.

D'altra banda, també tenim les plantilles, és aquí on hi ha tota la part de codi, és a dir, el JSP. En aquests arxius es recuperaran als diferents assets amb els seus respectius atributs i s'inclouran on toqui dins del codi HTML. A més, en cas necessari podem incorporar part de codi Java i aquest serà l'encarregat de gestionar tota la lògica.

### 3.2.3. *Funcionament*

Web CenterSites presenta una interfície web intuïtiva i fàcil de gestionar. Es divideix en diverses vistes o interfícies depenent de la funcionalitat a exercir.

- Interfície WEM Admin
- Interfície Administració
- Interfície Contribuïdora

#### 1. WEM Admin

La interfície WEM Admin és on es creen els Sites e Usuaris, també es gestionen les interfícies que el site té activades (Administració i Contribució).

#### 2. Vista administració

Des d'aquesta vista es podrà gestionar tot el contingut del portal i publicar continguts. Aquesta és la vista dedicada principalment a l'usuari desenvolupador, es pot gestionar tot el site permetent-ho entre altres funcions a creació de plantilles, publicació de continguts, editors d'atributs, etc.

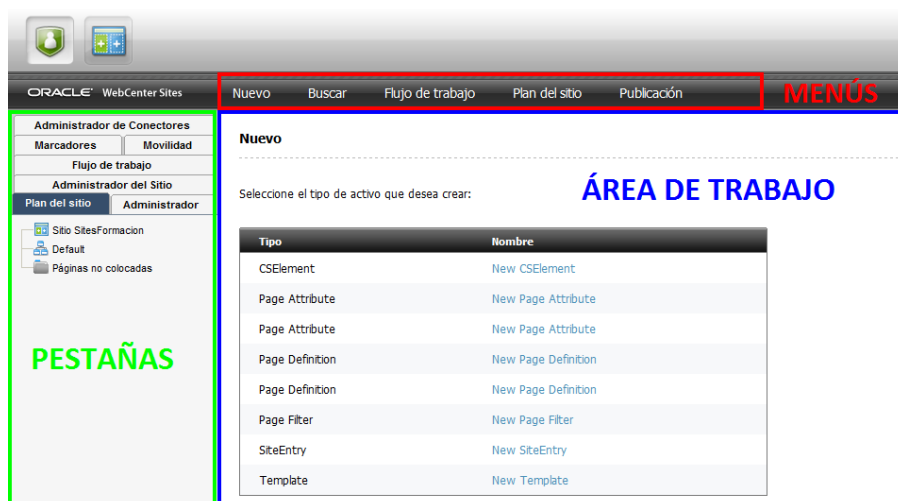
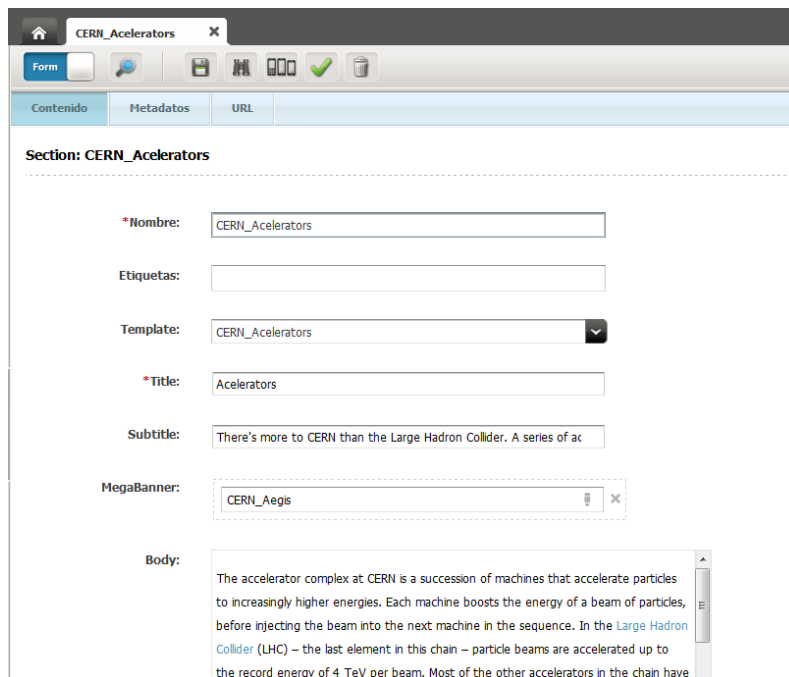


Figura 1. Vista administrador WCS

### 3. Vista contribució

Aquesta vista ens permet gestionar tot el contingut del portal, a continuació es mostra un exemple en mode formulari.



The screenshot shows a web browser window with the title 'CERN\_Acelerators'. The browser's address bar and navigation icons are visible. Below the browser window, there is a form titled 'Section: CERN\_Acelerators'. The form contains several fields: 'Nombre' (CERN\_Acelerators), 'Etiquetas' (empty), 'Template' (CERN\_Acelerators), '\* Title' (Acelerators), 'Subtitle' (There's more to CERN than the Large Hadron Collider. A series of ac), 'MegaBanner' (CERN\_Aegis), and 'Body' (The accelerator complex at CERN is a succession of machines that accelerate particles to increasingly higher energies. Each machine boosts the energy of a beam of particles, before injecting the beam into the next machine in the sequence. In the [Large Hadron Collider \(LHC\)](#) – the last element in this chain – particle beams are accelerated up to the record energy of 4 TeV per beam. Most of the other accelerators in the chain have

**Figura 2.** Vista contribució WCS

#### 3.2.4. Avantatges i inconvenients

A continuació es descriuen diferents avantatges i inconvenients d'Oracle WebCenter Sites.

Avantatges:

1. Gestió de contingut empresarial: WebCenter Sites proporciona una solució completa per a la gestió de contingut empresarial, la qual cosa permet a les organitzacions crear, gestionar i lliurar contingut de manera efectiva en diversos canals.
2. Personalització i segmentació: La plataforma ofereix eines poderoses per a la personalització i segmentació de contingut, la qual cosa permet a les organitzacions adaptar el contingut a les necessitats i preferències específiques de cada usuari.
3. Escalabilitat i rendiment: WebCenter Sites és altament escalable i pot gestionar grans volums de contingut i trànsit d'usuaris. Proporciona un rendiment robust i de confiança, la qual cosa garanteix una experiència d'usuari òptima fins i tot en entorns d'alta demanda.
4. Experiències digitals riques: La plataforma permet la creació d'experiències digitals riques i atractives mitjançant la incorporació de contingut multimèdia, la

personalització basada en el comportament de l'usuari i el lliurament en diferents dispositius i canals.

Inconvenients:

1. Complexitat d'implementació: La implementació de WebCenter Sites pot ser complexa i requerir coneixements tècnics avançats. Configurar i personalitzar la plataforma pot portar temps i recursos significatius.
2. Corba d'aprenentatge: A causa del seu ampli conjunt de característiques i funcionalitats, pot haver-hi una corba d'aprenentatge pronunciada per als usuaris i administradors nous. La formació i el temps d'adaptació poden ser necessaris per a aprofitar al màxim la plataforma.
3. Costos associats: Oracle WebCenter Sites és una solució empresarial i, com a tal, pot tenir costos significatius associats. Això inclou llicències de programari, suport tècnic, personal especialitzat i maquinari adequat.
4. Requisits d'infraestructura: La implementació de WebCenter Sites pot requerir una infraestructura de tecnologia robusta i escalable, la qual cosa implica inversions addicionals en maquinari, emmagatzematge i recursos de xarxa.

### **3.3. WordPress**

WordPress és un sistema de gestió de continguts (CMS) que permet als usuaris crear i administrar fàcilment llocs web dinàmics i funcionals. Va ser llançat per primera vegada en 2003 i des de llavors s'ha convertit en un dels CMS més populars del món, utilitzat per milions de llocs web.

Un dels principals avantatges de WordPress és la seva facilitat d'ús, inclús els usuaris sense coneixements tècnics poden crear i administrar llocs webs de manera senzilla gràcies a la seva interfície d'usuari intuïtiva i la gran quantitat de plugins i temes disponibles.

A més a més, WordPress és altament personalitzable i escalable, el que ho fa més adequat per llocs web de qualsevol mida i complexitat. Els usuaris poden personalitzar fàcilment l'aspecte i la funcionalitat dels seus llocs webs fent ús de codi predefinit o poden crear el seu propi codi personalitzat.

D'altra banda, WordPress també és altament compatible amb SEO<sup>3</sup>, el que ho fa ideal per als que desitgin millorar la seva visibilitat en els motors de recerca.

Per concloure, podem afirmar que és una plataforma CMS popular i altament flexible, ideal per als usuaris de tots els nivells d'habilitat que busquin crear llocs webs dinàmics i personalitzats.

---

<sup>3</sup> Search Engine Optimization

### **3.3.1. Evolució de Wordpress**

WordPress va ser creat en 2003 per Matt Mullenweg i Mike Little com un projecte derivat d'un altre sistema de blogging anomenat b2/\*cafelog. Des de llavors, ha evolucionat i ha crescut fins a convertir-se en el CMS que coneixem avui.

En els seus primers anys, WordPress era principalment utilitzat com una plataforma de blogging, permetent als usuaris crear i publicar contingut en línia de manera senzilla i eficient. No obstant això, amb el temps, s'ha convertit en molt més que això, i s'ha convertit en una plataforma molt versàtil que s'utilitza per a crear tot tipus de llocs web, des de botigues en línia fins a portafolis d'artistes.

En 2004, la versió 1.2 de WordPress va ser llançada al públic, i va ser la primera a incorporar el sistema de plugins que permet als usuaris afegir noves funcionalitats al sistema sense haver de modificar el codi font. Això va permetre que la comunitat de desenvolupadors s'involucrés en la creació de noves eines i característiques per a WordPress, la qual cosa ha ajudat al fet que creixi encara més.

La versió 2.0 de WordPress, llançada en 2005, va introduir la idea dels temes, que permeten als usuaris canviar l'aparença i la funcionalitat del seu lloc web amb facilitat. Aquesta característica ha estat fonamental per a l'èxit de WordPress, ja que ha permès que els usuaris personalitzin el seu lloc web sense haver d'aprendre a programar.

En 2008, es va llançar WordPress 2.5, que va ser una revisió important del CMS, amb una interfície d'usuari completament redissenyada i noves característiques com la galeria de mitjans i la possibilitat d'instal·lar temes i plugins directament des del panell d'administració.

Amb el temps, WordPress ha continuat evolucionant i millorant. La versió 3.0, llançada en 2010, va introduir la idea de les xarxes multilloc, que permet als usuaris crear i gestionar múltiples llocs web des d'una única instal·lació de WordPress.

En 2015, es va llançar WordPress 4.2, que va introduir l'opció d'usar emoticones en els continguts i en els comentaris. Aquesta va ser una petita, però divertida característica que va ajudar al fet que WordPress continués sent rellevant i atractiu per als usuaris.

En els últims anys, s'han centrat en millorar la seguretat i la velocitat del seu sistema, així com a fer-ho més accessible per als usuaris amb discapacitats visuals i altres impediments. També ha millorat la seva integració amb altres eines en línia, com les xarxes socials i els serveis de correu electrònic.

Avui dia, WordPress és utilitzat per milions de persones a tot el món. És una plataforma extremadament versàtil i fàcil d'utilitzar que ha permès a individus, organitzacions i empreses de totes les grandàries crear llocs web professionals i atractius sense haver de gastar grans quantitats de diners en programadors i dissenyadors web.

### **3.3.2. Estructura**

L'estructura de WordPress es divideix en tres parts principals:

- Base de dades: WordPress utilitza una base de dades MySQL per a emmagatzemar i recuperar contingut i configuracions del lloc web.
- Arxius del nucli: Els arxius principals de WordPress, incloent-hi els arxius PHP i els arxius de configuració, s'emmagatzemen en una carpeta anomenada "wp-admin" i "wp-includes" en el servidor web.

- Contingut del lloc web: El contingut del lloc web, com les publicacions i pàgines, s'emmagatzema en la base de dades MySQL i es mostra en la part frontal del lloc web a través dels arxius de tema i complements.

L'estructura dels arxius de tema i complements es basa en la jerarquia d'arxius, cosa que significa que WordPress busca arxius específics en un ordre predefinit per a construir i mostrar el contingut del lloc web. Els arxius de tema són responsables de l'aparença del lloc web i els complements agreguen funcionalitat addicional al lloc web.

En resum, WordPress utilitza una estructura modular, en la qual els arxius del nucli, els arxius de tema i els complements es combinen per a crear un lloc web funcional i personalitzat.

### **3.3.3. Avantatges i inconvenients**

Encara que WordPress té molts avantatges, també presenta alguns inconvenients. A continuació, explorarem tots dos aspectes detalladament.

Avantatges:

1. Fàcil d'usar: La facilitat d'ús és una dels principals avantatges de WordPress. Fins i tot si no tens experiència prèvia en disseny o desenvolupament web, pots crear un lloc web en WordPress. La plataforma té una interfície intuïtiva i fàcil d'usar que et permet crear i administrar contingut sense necessitat d'habilitats tècniques.
2. Àmplia selecció de plantilles: WordPress compta amb una àmplia selecció de plantilles (temes) gratuïtes i de pagament. Les plantilles són una excel·lent manera de personalitzar l'aspecte del teu lloc web sense haver d'escriure codi. A més, moltes plantilles són altament personalitzables, cosa que significa que pots ajustar-les per a adaptar-les a les teves necessitats.
3. Àmplia selecció de plugins: Els plugins són una característica clau de WordPress. Et permeten agregar funcions addicionals al teu lloc web, com a formularis de contacte, optimització de motors de cerca, integració amb xarxes socials, etc. Hi ha milers de plugins disponibles en la plataforma, cosa que significa que pots trobar una solució per a gairebé qualsevol necessitat.
4. SEO-friendly: WordPress és altament SEO-friendly (amigable amb els motors de cerca), cosa que significa que està dissenyat per a ajudar-te a millorar el rànquing del teu lloc web en els resultats de cerca de Google i altres motors de cerca. Hi ha varis plugins de SEO disponibles per a WordPress que t'ajudaran a optimitzar el teu lloc web per a les paraules clau rellevants.
5. Comunitat activa: WordPress té una comunitat molt activa de desenvolupadors, dissenyadors i usuaris. Hi ha innombrables blogs, fòrums i grups de discussió dedicats a la plataforma, cosa que significa que sempre hi ha algú disponible per a ajudar-te si tens preguntes o problemes.

Inconvenients:

1. **Seguretat:** WordPress és un dels CMS més populars del món, cosa que significa que és un objectiu atractiu per als hackers. Si no prens mesures de seguretat adequades, el teu lloc web podria estar en risc de ser piratejat. Afortunadament, hi ha moltes mesures que pots prendre per a protegir el teu lloc web, com instal·lar un plugin de seguretat, mantenir actualitzat WordPress i els plugins, i fer còpies de seguretat regulars del teu lloc web.
2. **Actualitzacions freqüents:** Encara que les actualitzacions són importants per a mantenir el teu lloc web segur i funcionant sense problemes, les actualitzacions freqüents de WordPress i els plugins poden ser un inconvenient. Cada actualització pot causar problemes de compatibilitat amb uns altres plugins o temes, cosa que significa que hauràs de solucionar aquests problemes manualment.
3. **Rendiment:** Si no es configura correctament, WordPress pot ser lent i poc eficient en termes de rendiment. Això pot ser un problema si tens un lloc web amb molt de trànsit o si necessites una velocitat de càrrega ràpida per als teus visitants. No obstant això, existeixen diverses solucions per a optimitzar el rendiment de WordPress, com l'ús de serveis d'allotjament optimitzats per a WordPress, l'optimització d'imatges i la utilització de plugins de caixet.
4. **Personalització:** Encara que WordPress ofereix una àmplia selecció de plantilles i plugins per a personalitzar el teu lloc web, si vols personalitzar-lo més enllà de les possibilitats que ofereix pot resultar molt complex.

### **3.4. Elecció d'un CMS**

Una vegada tenim clar que voler triar un sistema de gestió de continguts per a la nostra pàgina web, hem de decidir quin programari triem. Des del meu punt de vista, hi ha quatre criteris de selecció d'eines CMS:

- **Cobertura de requisits funcionals**

Un CMS permet la gestió de pàgines i continguts facilitant l'edició, publicacions d'aquests. També sobre altres aspectes necessaris en la gestió d'una web sigui pública o privada.

- **Característiques generals i diferencials**

Cada CMS té unes característiques comunes i unes altres diferents. Dins de les característiques generals es valora la qualitat o la cobertura.

- **Evolució del producte**

És important en l'elecció del producte el suport del fabricant i l'aposta que realitza pel que fa a actualitzacions, suport i garantia.

- **Llicència i preu**

Hem de destacar que com no tindrem mai les mateixes característiques el preus no son comparables literalment i sempre depèn de decisions i negociacions particulars amb el fabricant.

## 4. Part pràctica

Per poder dur a terme la part pràctica d'aquest treball fi de grau i, desenvolupar tots els coneixements esmentats anteriorment es realitzaran dos llocs webs, amb la màxima similitud possible, fetes des de dos CMS diferents, Oracle Web Center Sites i Wordpress.

### 4.1. Cas d'ús

La finalitat d'aquests dos llocs webs es mostrar diferents resultats obtinguts del projecte [Actua](#). Gràcies a l'API proporcionada pels investigadors obtenim les diferents dades en format JSON<sup>4</sup> i, el nostre objectiu és llegir el proporcionat i mostrar les diferents dades de diverses formes.

Per a poder-ho implementar, farem servir un JavaScript i un CSS públic i la nostra funció es desenvolupar la part BackEnd. Aquesta part consisteix en arribar a la API proporcionada, llegir les dades i una vegada tinguem accés directe a elles, mostrar-les de diferents maneres, com per exemple, mostrar una taula amb les últimes 5 dades d'una escola, aquestes dades poden ser la temperatura, humitat, pressió i concentració, a banda de tindre un enllaç directe a les dades dels últims set dies. També podem mostrar diferents mòduls amb altres informacions respecte a les escoles recollides en l'API.

#### 4.1.3. Informació sobre l'API

L'API proporcionada pels membres del projecte Actua a la Universitat Rovira i Virgili i que ens proporciona diferents dades, té diversos paràmetres que podem combinar en funció de la informació que vulguem obtenir. Els paràmetres són els següents:

- recording\_id: aquest paràmetre és l'indicador de l'escola, cada escola es identificada pel seu propi ID. Aquest ID és de tipus numèric. Alguns exemples són:
  - o 25 correspon a una aula d'una escola de Tivenys
  - o 74 correspon a una aula d'una escola de Barcelona
  - o 82 correspon a una aula d'una escola de Molins de Rei
  - o 98 correspon a una aula d'una escola de Valls
  - o 109 correspon a una aula d'una escola de Cambrils
  - o 130 correspon a una aula d'una escola de Roda de Berà
- parameter\_id: el següent paràmetre fa referència a la dada ambiental que volem aconseguir.
  - o TEMP: temperatura (°C)
  - o HUM: humitat relativa (%)
  - o PRES: pressió baromètrica (hPa)
  - o CO2: concentració de CO2 (ppm)
  - o PM1: concentració de partícules PM1 (ug/m3)
  - o PM25: concentració de partícules PM2.5 (ug/m3)
  - o PM10: concentració de partícules PM10 (ug/m3)
  - o NOISE: pressió sonora (dB)
  - o MOVEMENT: quantitat de moviment (Índex)
  - o LLUM: lluminositat (lux)
  - o INDEX\_UV: radiació UV (Índex)

---

<sup>4</sup> JavaScript Object Notation

- `init_date`: Data d'inici del rang per obtenir les dades. Format YYYY-MM-DD
- `end_date`: Data de finalització del rang per obtenir les dades. Format YYYY-MM-DD

Un exemple d'ús d'aquesta API és el següent:

[https://actua-urv.cat/app/api/get\\_contextual\\_data.php?recording\\_id=25&parameter\\_id=TEMP&init\\_date=2023-05-14&end\\_date=2023-05-21](https://actua-urv.cat/app/api/get_contextual_data.php?recording_id=25&parameter_id=TEMP&init_date=2023-05-14&end_date=2023-05-21)

En aquest cas s'indica un ID referent a l'escola de Tivenys, volem obtenir dades sobre la temperatura amb data d'inici el 14 de maig de 2023 i data de finalització el 21 de maig de 2023.

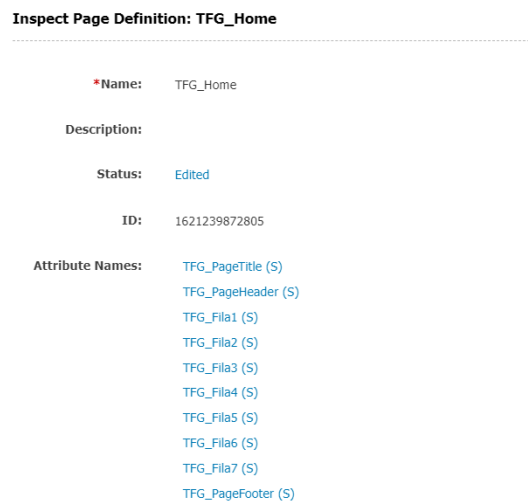
## 4.2. Oracle WebCenter Sites

Tal com s'ha explicat anteriorment, Oracle WebCenter Sites és un sistema de gestió de continguts, amb això aconseguim que una vegada estigui desenvolupada la part tècnica, qualsevol persona pot gestionar els diferents continguts sense cap coneixement especialitzat en desenvolupament web. A continuació, entrarem en el detall tècnic per veure com és el seu funcionament i, per acabar, una breu guia per tal de gestionar els continguts.

### 4.2.1. Desenvolupament tècnic

Per començar el desenvolupament d'un lloc web fent servir el CMS Oracle WebCenter Sites hem de crear les diferents famílies flexibles amb els seus respectius atributs.

Primer de tot, es crea la definició de pàgina i, cada pàgina tindrà un títol, una capçalera, set files per a poder introduir contingut i per últim, un peu de pàgina.



**Figura 3.** Definició de pàgina

Una vegada tenim la definició creada, passem a crear la seva plantilla. En aquesta establirem l'estructura d'una pàgina web HTML i dins d'aquí, recuperarem cadascun dels atributs.

```

<html>
  <head>

  <body class="row-flex">
    <main class="flex-shrink-0">
      <!-- Header-->
      <c:if test="${!empty page.TFG PageHeader}">
      <!-- //Header -->
      <!-- Modulos-->
      <c:if test="${!empty page.TFG Fila1}">
      <c:if test="${!empty page.TFG Fila2}">
      <c:if test="${!empty page.TFG Fila3}">
      <c:if test="${!empty page.TFG Fila4}">
      <c:if test="${!empty page.TFG Fila5}">
      <c:if test="${!empty page.TFG Fila6}">
      <c:if test="${!empty page.TFG Fila7}">
      <!-- //Modulos -->
      <!-- Footer-->
      <c:if test="${!empty page.TFG PageFooter}">
      <!-- //Footer -->
    </main>
  </body>
</html>

```

**Figura 4.** Estructura d'una web HTML i crida als atributs

Dins d'aquest element també afegirem el títol principal del lloc web i farem la crida al JavaScript i CSS públics que utilitzarem.

```

<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
  <meta name="description" content="" />
  <meta name="author" content="" />
  <title>Actua - TFG Esther</title>
  <!-- Favicon-->
  <link rel="icon" type="image/x-icon" href="assets/favicon.ico" />
  <!-- Bootstrap icons-->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css" rel="stylesheet" />
  <!-- Core theme CSS (includes Bootstrap)-->
  <link href="https://startbootstrap.github.io/startbootstrap-modern-business/css/styles.css" rel="stylesheet" />
  <!-- Bootstrap core JS-->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
  <!-- Core theme JS-->
  <script src="https://startbootstrap.github.io/startbootstrap-modern-business/js/scripts.js"></script>
</head>

```

**Figura 5.** Head HTML i crida al JavaScript i CSS

A continuació, es mostra un exemple de com es la crida a un altre objecte, en aquest cas, si tenim contingut a la capçalera, farem la crida a la seva plantilla corresponent.

```

<c:if test="${!empty page.TFG_PageHeader}">
  <render:calltemplate tname="TFG_Header" c="TFG_Module_P" cid='${page.TFG_PageHeader}' site='${args.site}' ttype="Template" />
</c:if>

```

**Figura 6.** Crida a la capçalera

Quan ja està la pàgina amb el seu format i totes les crides als diferents elements feta, passem a cadascun dels components que tindrem.

Primer de tot, s'han de crear les diferents famílies flexibles, nosaltres en tindrem

dos: *Content* i *Module*. Dins de cadascuna d'aquestes hi han els seus subtipus. En el cas dels *Contents* hi trobem els enllaços i les imatges, que després utilitzarem en els mòduls. Dins de la família flexible *Modules* trobarem els subtipus de cada mòdul, com la capçalera, el peu i els diferents mòduls creats.

Seguidament, analitzarem el mòdul de la capçalera. Aquest mòdul consta de dos atributs, el títol de la capçalera i un atribut múltiple dels enllaços d'aquesta. L'últim atribut és de tipus asset, pertany a la família flexible *Content* i el seu subtipus és *ContentLink*.

**Inspect TFG\_Module\_CD: TFG\_ModuleHeader**

**\*Name:** TFG\_ModuleHeader

**Description:**

**Status:** Edited

**ID:** 1621239872837

**Parent Definitions:** (none)

**Attribute Names:** TFG\_ModuleHeaderTitle (S)  
TFG\_ModuleHeaderLink (M)

**TFG\_Module\_A: TFG\_ModuleHeaderLink**

**\*Name:** TFG\_ModuleHeaderLink

**Description:** Enlaces del header

**Status:** Created

**ID:** 1621239875280

**Site:** Site exclusivo TFG Esther

**Attribute Type:** asset

**Asset Type:** TFG\_Content\_P

**Asset Subtypes:** TFG\_ContentLink

**Figura 7.** Mòdul de la capçalera

**Figura 8.** Atribut enllaços de la capçalera

El subtipus *ContentLink* consta de dos atributs, un títol obligatori i el seu enllaç.

**Inspect TFG\_Content\_CD: TFG\_ContentLink**

**\*Name:** TFG\_ContentLink

**Description:** Link

**Status:** Edited

**ID:** 1621239875253

**Parent Definitions:** (none)

**Attribute Names:** \*TFG\_ContentLinkTitle (S)  
TFG\_ContentLinkHref (S)

**Figura 9.** Atributs d'un enllaç

Per acabar amb la capçalera, ens endinsem en el codi per poder entendre com es recuperen cadascun dels diferents atributs que hem esmentat fins ara i com es

col·loquen a la seva posició correcta.

Seguint l'estructura bàsica de l'HTML, una capçalera sempre consta d'una navegació amb una llista dins. Fem servir el llenguatge JSP per tal de poder fer un bucle i pintar tants enllaços com s'hagin contribuït.

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container px-5">
    <a class="navbar-brand" href="http://localhost:7003/sites/samples/homeEsther">${headerTFG.TFG_ModuleHeaderTitle}</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
      aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
        <:forEach items="${headerTFG.TFG_ModuleHeaderLink}" var="itemEnlace">
          <vf:getattributes name="enlace" assettype="TFG_Content_P" assetid="${itemEnlace.id}" attributes="TFG_ContentLinkTitle,TFG_ContentLinkHref"/>
          <li class="nav-item"><a class="nav-link" href="${enlace.TFG_ContentLinkHref}">${enlace.TFG_ContentLinkTitle}</a></li>
        </forEach>
      </ul>
    </div>
  </div>
</nav>
```

**Figura 10.** Codi de la capçalera

Per continuar amb els elements bàsics d'una pàgina, continuarem amb el seu peu. Aquest nou element és molt semblant a la capçalera, però amb les seves pròpies necessitats que s'han identificat.

Com ja s'ha esmentat anteriorment, el peu forma part de la família flexible de *Modules* i està format per un títol i els seus enllaços. De la mateixa manera que els enllaços de la capçalera, aquest atribut també és un *Asset* de tipus *Content* i de subtipus *ContentLink*.

**Inspect TFG\_Module\_CD: TFG\_ModuleFooter**

<b>*Name:</b>	TFG_ModuleFooter
<b>Description:</b>	
<b>Status:</b>	Edited
<b>ID:</b>	1621239872607
<b>Parent Definitions:</b>	(none)
<b>Attribute Names:</b>	TFG_ModuleFooterTitulo (S) TFG_ModuleFooterLink (M)

**TFG\_Module\_A: TFG\_ModuleFooterLink**

<b>*Name:</b>	TFG_ModuleFooterLink
<b>Description:</b>	Enlaces del footer
<b>Status:</b>	Edited
<b>ID:</b>	1621239877893
<b>Site:</b>	Site exclusivo TFG Esther
<b>Attribute Type:</b>	asset
<b>Asset Type:</b>	TFG_Content_P
<b>Asset Subtypes:</b>	TFG_ContentLink

**Figura 11.** Mòdul del peu

**Figura 12.** Atribut enllaços del peu

El codi del peu de pàgina segueix l'estructura principal d'HTML, nosaltres hi incorporem codi JSP per poder recuperar cadascun dels atributs. En primer lloc, pintem el títol del peu de pàgina i, en segon lloc, fent servir un bucle, recorrem cadascun dels enllaços que s'hagin contribuït.

```
<footer class="bg-dark py-4 mt-auto">
  <div class="container px-5">
    <div class="row align-items-center justify-content-between flex-column flex-sm-row">
      <div class="col-auto"><div class="small m-0 text-white">${footer.TFG_ModuleFooterTitulo}</div></div>
      <div class="col-auto">
        <:if test="(!empty footer.TFG_ModuleFooterLink)">
          <:forEach items="${footer.TFG_ModuleFooterLink}" var="itemEnlace">
            <vf:getattributes name="enlace" assettype="TFG_Content_P" assetid="${itemEnlace.id}" attributes="TFG_ContentLinkTitle,TFG_ContentLinkHref"/>
            <a class="link-light small" href="${enlace.TFG_ContentLinkHref}">${enlace.TFG_ContentLinkTitle}</a>
          </forEach>
        </c:if>
      </div>
    </div>
  </div>
</footer>
```

**Figura 13.** Codi del peu

Una vegada tenim l'estructura de la pàgina, la capçalera i el peu continuarem creant els diferents mòduls que tindrem disponibles per a poder-los contribuir en qualsevol lloc del portal web.

El primer mòdul que s'ha volgut crear és un mòdul introductori al lloc web, on podrà aparèixer un títol i/o descripció, una imatge i la possibilitat d'incloure enllaços. Aquest mòdul rep el nom de *ModuleHeroBanner* i està format per quatre atributs. Un títol, una descripció, una imatge i uns enllaços. Cal destacar que igual que a la resta dels mòduls, cap dels atributs és de caràcter obligatori, podem decidir afegir-hi un contingut o no.

#### Inspect TFG\_Module\_CD: TFG\_ModuleHeroBanner

---

**\*Name:** TFG\_ModuleHeroBanner

**Description:**

**Status:** Created

**ID:** 1621239876006

**Parent Definitions:** (none)

**Attribute Names:** TFG\_ModuleTitulo (S)  
TFG\_ModuleDescripcion (S)  
TFG\_ModuleImagen (S)  
TFG\_ModuleEnlaces (M)

**Figura 11.** Mòdul Hero Banner

Els atributs de títol i descripció són de tipus *String*<sup>5</sup>, la imatge és un *Asset* de tipus *Content* i de subtipus *ContentImage* i, l'últim atribut que fa referència als enllaços també és un *Asset* però de subtipus *ContentLink*.

Continuarem explicant el subtipus *ContentImage*, aquest conté tres atributs: el títol de la imatge, la imatge (de caràcter obligatori) de tipus blob<sup>6</sup> i per últim un text alternatiu de la imatge.

#### Inspect TFG\_Content\_CD: TFG\_ContentImage

---

**\*Name:** TFG\_ContentImage

**Description:**

**Status:** Created

**ID:** 1621239872688

**Parent Definitions:** (none)

**Attribute Names:** TFG\_ContentImageTitle (S)  
\*TFG\_ContentImageImage (S)  
TFG\_ContentImageImageAlt (S)

---

<sup>5</sup> Seqüència de caràcters

<sup>6</sup> Binary Large Object

Figura 12. Atributs d'una imatge

Una vegada tenim clara l'estructura del mòdul que anteriorment hem anomenat *ModuleHeroBanner* passarem al seu funcionament a través del codi HTML i JSP.

```
<vf:getcontext args='c,cid,tid,sitefx,site,rendermode' />
<vf:getattributes name="hero" assettype="TFG_Module_P" assetid="${args.cid}" attributes="TFG_ModuleTitulo,TFG_ModuleDescripcion,TFG_ModuleImagen,TFG_ModuleEnlaces"/>
<header class="py-5">
  <div class="container px-5 my-5">
    <div class="row gx-5 align-items-center justify-content-center">
      <div class="col-lg-8 col-xl-7 col-xxl-6">
        <div class="my-5 text-center text-xl-start">
          <h1 class="display-5 fw-bolder mb-2">${hero.TFG_ModuleTitulo}</h1>
          <p class="lead fw-normal text-muted mb-4">${hero.TFG_ModuleDescripcion}</p>
          <div class="d-grid gap-3 d-sm-flex justify-content-sm-center justify-content-xl-start">
            <c:forEach items="${hero.TFG_ModuleEnlaces}" var="itemEnlace">
              <vf:getattributes name="enlace" assettype="TFG_Content_P" assetid="${itemEnlace.id}" attributes="TFG_ContentLinkTitle,TFG_ContentLinkHref"/>
              <a class="btn btn-primary btn-lg px-4 me-sm-3" href="${enlace.TFG_ContentLinkHref}">${enlace.TFG_ContentLinkTitle}</a>
            </c:forEach>
          </div>
        </div>
      </div>
    </div>
    <div class="col-xl-5 col-xxl-6 d-none d-xl-block text-center">
      <vf:getattributes name="imagen" assettype="TFG_Content_P" assetid="${hero.TFG_ModuleImagen}" attributes="TFG_ContentImageImage,TFG_ContentImageImageAlt"/>
      <render:getbloburl
        c="TFG_Content_P"
        field="TFG_ContentImageImage"
        cid="${hero.TFG_ModuleImagen}"
        outstr="theURL"/>
      
    </div>
  </div>
</header>
```

Figura 13. Codi del mòdul Hero Banner

Com es pot observar es tracta d'una estructura HTML i la nostra funció és recuperar els diferents atributs que posteriorment es contribuiran tantes vegades com vulguem dins del nostre portal web. Podem veure com el títol l'introduïm en una etiqueta h1 i la descripció dins de l'etiqueta de paràgraf. En segon lloc, tenim els enllaços, i com hem esmentat anteriorment podem afegir més d'un, és per aquest motiu que hem de recórrer un bucle per cadascun dels enllaços contribuïts. Com sabeu, l'atribut dels enllaços, és un *Asset* del subtipus *ContentLink* que té els seus propis atributs, per tant, també els haurem de recuperar. Aquest els inclourem dins d'on toca seguit l'estructura de l'HTML. Pintarem el seu enllaç, col·loquialment conegut com a *href* i el títol, aquest serà visible a simple al nostre lloc web.

Per acabar amb aquest mòdul, ens falta la imatge. Aquesta també és un tipus *Asset* i de subtipus *ContentImage*, per tant, igual que hem fet anteriorment amb els enllaços, hem de recuperar els seus propis atributs. En aquest cas, per obtenir la imatge hem d'obtenir l'URL del blob, una vegada ho tenim ja la podrem pintar.

Posteriorment, es crea un nou mòdul que ens donarà la possibilitat d'un títol, subtítol i d'afegir diferents elements, aquests els veurem més endavant. Aquest nou mòdul l'hem anomenat *Sections*.

#### Inspect TFG\_Module\_CD: TFG\_ModuleSections

---

**\*Name:** TFG\_ModuleSections

**Description:**

**Status:** Edited

**ID:** 1621239876431

**Parent Definitions:** (none)

**Attribute Names:** TFG\_ModuleTitulo (S)  
TFG\_ModuleDescripcion (S)  
TFG\_ModuleSections (M)

**Figura 14.** Mòdul Section

L'atribut *Sections* és un *Asset* de tipus *Content* i subtipus *ContentSection*.

#### TFG\_Module\_A: TFG\_ModuleSections

---

**\*Name:** TFG\_ModuleSections

**Description:** Secciones

**Status:** Created

**ID:** 1621239876465

**Site:** Site exclusivo TFG Esther

**Attribute Type:** asset

**Asset Type:** TFG\_Content\_P

**Asset Subtypes:** TFG\_ContentSection

**Figura 15.** Atribut Sections

A continuació analitzarem aquest subtipus de contingut, consta d'un títol, imatge, text alternatiu d'una imatge, títol de l'enllaç i per últim enllaç.

### Inspect TFG\_Content\_CD: TFG\_ContentSection

*Name:	TFG_ContentSection
Description:	
Status:	Edited
ID:	1621239876438
Parent Definitions:	(none)
Attribute Names:	TFG_ContentTitle (S) TFG_ContentImageImage (S) TFG_ContentImageImageAlt (S) TFG_ContentLinkTitle (S) TFG_ContentLinkHref (S)

Figura 116. Atributs de cada secció

Una vegada tenim l'estructura del mòdul definida, anem al seu codi, és aquí on es estructuren els diferents continguts.

```
<section class="py-5 bg-light">
  <div class="container px-5 my-5">
    <div class="text-center">
      <h2 class="fw-bolder">${seccion.TFG_ModuleTitulo}</h2>
      <p class="lead fw-normal text-muted mb-5">${seccion.TFG_ModuleDescripcion}</p>
    </div>
    <div class="row gx-5 row-cols-1 row-cols-sm-2 row-cols-xl-4 justify-content-center">
      <c:if test="{!empty seccion.TFG_ModuleSections}">
        <c:forEach items="{seccion.TFG_ModuleSections}" var="itemSection">
          <vf:getattributes name="section" assettype="TFG_Content_P" assetid="{itemSection.id}"
            attributes="TFG_ContentTitle,TFG_ContentImageImage,TFG_ContentImageImageAlt,TFG_ContentLinkTitle,TFG_ContentLinkHref"/>
          <div class="col mb-5 mb-5 mb-xl-0">
            <div class="text-center">
              <render:getbloburl
                c="TFG_Content_P"
                field="TFG_ContentImageImage"
                cid="{itemSection.id}"
                outstr="theURL"/>
              
              <h5 class="fw-bolder">${section.TFG_ContentTitle}</h5>
              <c:set var="urlJSON" value="{section.TFG_ContentLinkHref}" />
              <a href="{urlJSON}">${section.TFG_ContentLinkTitle}</a>
            </div>
          </div>
          <render:callelement elementname="tfG/lecturajson" scoped="global">
        </c:forEach>
      </c:if>
    </div>
  </div>
</section>
```

Figura 17. Estructura del codi HTML del mòdul Sections

Es pot observar com al principi de l'estructura HTML cridem al títol i a la descripció. Més endavant es fa un bucle per a poder recuperar les diverses seccions que s'afegeixen. Per començar amb cada secció recuperarem la imatge i el seu títol accessible i el col·locarem dins de la seva pròpia etiqueta d'HTML.

En segon lloc, recuperem el contingut de l'enllaç. En aquesta secció l'enllaç té la funció de mostrar les dades de l'API i per això sempre que el contribuïrem ha de complir les condicions següents: el paràmetre *init\_date* ha de ser: xxxxx i el paràmetre *end\_date*: yyyy. La funció del codi és reemplaçar la data inicial per 2023-01-01 i la data final per la data actual.

```

<%
String urlAPI = (String) pageContext.getAttribute("urlJSON");
Date date = new Date();
String fechaFin = "";
String fechaI = "2023-01-01";
Locale aLocale= new Locale("es", "ES");
DateFormat formatoFecha = new SimpleDateFormat("yyyy-MM-dd", aLocale);
fechaFin = formatoFecha.format(date);
if (urlAPI.contains("xxxxx") && urlAPI.contains("yyyyy")){
    urlAPI= urlAPI.replaceAll("yyyyy", fechaFin);
    urlAPI= urlAPI.replaceAll("xxxxx", fechaI);
}
%>

```

**Figura 18.** Codi Java per reemplaçar paràmetres

Per finalitzar amb aquesta secció, cridarem a un nou element de codi anomenat “lecturajson”. Aquest serà l’encarregat de pintar un enllaç on podrem veure les dades dels últims set dies i una taula amb les cinc últimes dades que tinguem.

```

<render:callelement elementname="tfg/lecturajson" scoped="global">
    <render:argument name="c" value="TFG_Content_P" />
    <render:argument name="urlAtr" value="{section.TFG_ContentLinkHref}" />
    <render:argument name="site" value="{args.site}" />
</render:callelement>

```

**Figura 19.** Crida a l’element lecturajson

Comencem l’element de codi reemplaçant l’enllaç de l’API, el paràmetre de la data final passarà a ser la data actual i el paràmetre de data actual correspondrà a la data de fa una setmana, és a dir, set dies anteriors a la data actual. D’aquesta manera podrem veure les dades dels últims set dies.

```

<%
String urlAPI = (String) pageContext.getAttribute("urlJSON");

Date date = new Date();
String fechaFin = "";
String fechaI = "";
Locale aLocale= new Locale("es", "ES");
DateFormat formatoFecha = new SimpleDateFormat("yyyy-MM-dd", aLocale);
fechaFin = formatoFecha.format(date);

Calendar c = Calendar.getInstance();
c.setTime(date);
c.add(Calendar.DATE, -7);

Date fechaIni = c.getTime();
fechaI = formatoFecha.format(fechaIni);
if (urlAPI.contains("xxxxx") && urlAPI.contains("yyyyy")){

    urlAPI= urlAPI.replaceAll("yyyyy", fechaFin);
    urlAPI= urlAPI.replaceAll("xxxxx", fechaI);

}

%>

<div class="fst-italic text-muted">
    <a href='<%=urlAPI%>'>Veure les dades dels últims 7 dies</a>
</div>

<%

```

**Figura 20.** Codi Java on es reemplaçen les dates per mostrar les dades dels últims 7 dies

A continuació, s'establirà una connexió amb aquesta última URL i llegirem el paràmetre anomenat: *parameter\_id*. Gràcies a aquest paràmetre podrem saber si es tracten de dades referents a la temperatura, humitat, pressió, etc. Aquesta dada serà emmagatzemada en una variable que utilitzarem més endavant.

```

URL url = new URL(urlAPI);
URLConnection uc = url.openConnection();
uc.connect();
String valueParam = "";
if (urlAPI.contains("TEMP")){
    valueParam = "Temperatura (°C)";
}
if (urlAPI.contains("HUM")){
    valueParam = "Humitat relativa (%)";
}
if (urlAPI.contains("PRES")){
    valueParam = "Pressió baromètrica (hPa)";
}

```

**Figura 21.** Reemplaçament dels identificadors dels paràmetres

Seguidament, llegirem tot el contingut que ens retorna l'API, en quedarem amb les últimes cinc dades i les mostrarem en una taula. La taula tindrà dues columnes: els valors de les dades i les seves dates. A la capçalera indicarem de quin paràmetre es tracta, això ho farem gràcies al paràmetre llegit anteriorment de la URL. Després recorrent un bucle pintarem les dades corresponents.

```

<table class="table">
  <thead>
    <tr class="table-primary">
      <th scope="col"><%=valueParam%></th>
      <th scope="col">Data</th>
    </tr>
  </thead>
  <tbody>
    <%
      for (int i = arrayObject.length()-5; i < arrayObject.length(); i++) {
        JSONObject object = (JSONObject) arrayObject.get(i);
        String x = object.get("x").toString(); //fecha
        String [] partsTemp = x.split("\\s+");
        x = partsTemp[1];
        String y = object.get("y").toString(); //temp
        //out.println("tiempo: "+y+" temp: "+x);
        %>
        <tr>
          <td><%=y%></td>
          <td><%=x%></td>
        </tr>
      <%
    }
    %>
  </tbody>
</table>

```

**Figura 22.** Codi HTML i Java per obtenir les dades pertinents i elaborar una taula

El següent mòdul que crearem l'anomenarem *Features*, característiques en anglès. Aquest només contindrà dos atributs: un títol i les seves característiques.

#### Inspect TFG\_Module\_CD: TFG\_ModuleFeatures

**\*Name:** TFG\_ModuleFeatures  
**Description:**  
**Status:** Edited  
**ID:** 1621239877206  
**Parent Definitions:** (none)  
**Attribute Names:** TFG\_ModuleTitulo (S)  
TFG\_ModuleFeatures (M)

**Figura 23.** Mòdul Features

#### TFG\_Module\_A: TFG\_ModuleFeatures

**\*Name:** TFG\_ModuleFeatures  
**Description:** Features  
**Status:** Created  
**ID:** 1621239877212  
**Site:** Site exclusivo TFG Esther  
**Attribute Type:** asset  
**Asset Type:** TFG\_Content\_P  
**Asset Subtypes:** TFG\_ContentFeatures

**Figura 24.** Atribut Feature

Igual que en mòduls anteriors, les diferents característiques que tindrem seran un *Asset* de tipus contingut i subtipus *Content Features*.

#### Inspect TFG\_Content\_CD: TFG\_ContentFeatures

**\*Name:** TFG\_ContentFeatures  
**Description:** Featues  
**Status:** Edited  
**ID:** 1621239877177  
**Parent Definitions:** (none)  
**Attribute Names:** TFG\_ContentTitle (S)  
TFG\_ContentDescripcion (S)  
TFG\_ContentImageImage (S)  
TFG\_ContentImageImageAlt (S)  
TFG\_ContenLinkTitle (S)  
TFG\_ContenLinkHref (S)

**Figura 25.** Atributs del contingut Feature

El subtipus de contingut *Content Features* consta del següents atributs: un títol, descripció, imatge, títol accessible de la imatge, títol de l'enllaç i el propi enllaç.

Una vegada l'estructura del mòdul està creada, hem de continuar pel pas més important, el codi. És en aquest element on es decideix com estructurarem el nostre contingut, gràcies a una estructura HTML.

```

<section class="py-5" id="features">
  <div class="container px-5 my-5">
    <div class="row gx-5">
      <div class="col-lg-4 mb-5 mb-lg-0"><h2 class="fw-bold">${features.TFG_ModuleTitulo}</h2</div>
      <div class="col-lg-8">
        <div class="row gx-5 row-cols-1 row-cols-md-2">
          <:if test="${!empty features.TFG_ModuleFeatures}">
            <:forEach items="${features.TFG_ModuleFeatures}" var="itemFeature">
              <:forEach attributes="TFG_ContentTitle,TFG_ContentDescripcion,TFG_ContentImageImage,TFG_ContentImageImageAlt,TFG_ContentLinkTitle,TFG_ContentLinkHref">
                <render:getTitleUrl
                  c="${TFG_Content_P"
                  field="TFG_ContentImageImage"
                  cid="${itemFeature.id}"
                  outstr="theURL"/>
                <div class="col mb-5 h-100">
                  <div class="bg-gradient text-white rounded-3 mb-3">
                    
                  </div>
                  <h2 class="h5">${feature.TFG_ContentTitle}</h2>
                  <p class="mb-0">${feature.TFG_ContentDescripcion}</p>
                  <:set var="urlJSON" value="${feature.TFG_ContentLinkHref}"/>
                  <
                    String urlAPI = (String) pageContext.getAttribute("urlJSON");
                    Date date = new Date();
                    String fechaFin = "";
                    String fechaI = "2023-01-01";
                    Locale aLocale = new Locale("es", "ES");
                    DateFormat formatoFecha = new SimpleDateFormat("yyyy-MM-dd", aLocale);
                    fechaFin = formatoFecha.format(date);
                    if (urlAPI.contains("xxxxx") && urlAPI.contains("yyyyy")){
                      urlAPI= urlAPI.replaceAll("yyyyy", fechaFin);
                      urlAPI= urlAPI.replaceAll("xxxxx", fechaI);
                    }
                  >
                  <div class="fst-italic text-muted">
                    <a href="${urlAPI}">${feature.TFG_ContentLinkTitle}</a>
                  </div>
                </div>
              </:forEach>
            </:if>
          </div>
        </div>
      </div>
    </div>
  </section>

```

Figura 26. Codi del mòdul Features

En la imatge anterior es mostra l'estructura HTML d'aquest mòdul, codi JSP per recuperar els diferents atributs i codi Java per introduir la lògica necessària.

L'atribut únic del títol es pinta amb una etiqueta d'HTML *h2* i per recuperar cadascuna de les seccions farem servir un bucle de JSP: *forEach*. Aquest bucle es recorrerà tantes vegades com característiques es contribueixin. És aquí on pintarem els atributs de cada característica i reemplaçarem la data d'inici per "2023-01-01" i la data final per l'actual.

El pròxim mòdul rep el nom *School*, en català, escola. És anomenat així perquè està pensat per a mostrar dades d'una determinada escola. Els seus atributs són: títol, descripció i un enllaç a l'API.

Inspect TFG\_Module\_CD: TFG\_ModuleSchool

*Name:	TFG_ModuleSchool
Description:	
Status:	Edited
ID:	1621239886282
Parent Definitions:	(none)
Attribute Names:	TFG_ModuleTitulo (S) TFG_ModuleDescripcion (S) TFG_ModuleLinkApi (S)

Figura 27. Mòdul School

L'enllaç a l'API s'haurà de contribuir de manera correcta per tal de el mòdul faci la seva funció. El paràmetre *parameter\_id* ha de ser igual a *p\_id*, el *init\_date* ha de ser *xxxxx* i per últim, el paràmetre *end\_date* ha de ser *yyyyy*.

Seguidament, analitzarem el codi i entendrem el perquè d'aquesta singularitat en l'enllaç.

```

<section class="py-5">
  <div class="container px-5 my-5">
    <div class="row gx-5 align-items-center justify-content-center">
      <div class="col-lg-8 col-xl-7 col-xxl-6">
        <div class="my-5 text-center text-xl-start">
          <h1 class="display-5 fw-bolder mb-2">${school.TFG_ModuleTitulo}</h1>
          <p class="lead fw-normal text-muted mb-4">${school.TFG_ModuleDescripcion}</p>
          <div class="fst-italic text-muted">
            <a href="<%=urlTemp%>">Veure la temperatura dels ultims 7 dies</a>
          </div>
          <div class="fst-italic text-muted">
            <a href="<%=urlHum%>">Veure la humitat dels ultims 7 dies</a>
          </div>
          <div class="fst-italic text-muted">
            <a href="<%=urlPres%>">Veure la pressio dels ultims 7 dies</a>
          </div>
          <div class="fst-italic text-muted">
            <a href="<%=urlCO2%>">Veure la concentracio de CO2 dels ultims 7 dies</a>
          </div>
        </div>
      </div>
      <div class="col-xl-5 col-xxl-6 d-none d-xl-block text-center">
        <table class="table">
          <thead>
            <tr class="table-primary">
              <th scope="col"></th>
              <th scope="col">Temperatura</th>
              <th scope="col">Humitat</th>
              <th scope="col">Pressio</th>
              <th scope="col">Concentracio</th>
            </tr>
          </thead>
          <tbody>
            <tr>
              <th scope="row"><%=hora%></th>
              <td><%=tempTable%></td>
              <td><%=humTable%></td>
              <td><%=presTable%></td>
              <td><%=co2Table%></td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </div>
</section>

```

**Figura 28.** Estructura HTML del mòdul School

En la figura anterior es mostra l'estructura HTML del mòdul *School*, on es pot observar com hi haurà un títol, una descripció, quatre enllaços amb les dades de la temperatura, humitat, pressió baromètrica i concentració de CO2 dels últims set dies. També apareix una taula on les seves files són la data de les dades, la temperatura, humitat, pressió i concentració.

Per poder fer possible l'estructura esmentada necessitem aplicar lògica en llenguatge de programació Java. En primer lloc, amb l'enllaç proporcionat pel contribuïdor de manera correcta, inclourem la data d'inici a 1 de gener de 2023 i com a data final, la data actual. En segon lloc, substituïrem l'identificador del paràmetre pels propis identificadors de la temperatura, humitat, pressió i concentració CO2, creant així quatre enllaços diferents amb els quals establirem una connexió HTTP.

```
<c:set var="urlApi" value="${school.TFG_ModuleLinkApi}"/>
<#
String urlAPI = (String) pageContext.getAttribute("urlApi");
Date date = new Date();
String fechaFin = "";
String fechaI = "";
Locale aLocale= new Locale("es", "ES");
DateFormat formatoFecha = new SimpleDateFormat("yyyy-MM-dd", aLocale);
fechaFin = formatoFecha.format(date);
Calendar c = Calendar.getInstance();
c.setTime(date);
c.add(Calendar.DATE, -7);

Date fechaIni = c.getTime();
fechaI = formatoFecha.format(fechaIni);

if (urlAPI.contains("xxxxxx") && urlAPI.contains("yyyyyy")){
    urlAPI= urlAPI.replaceAll("yyyyyy", fechaFin);
    urlAPI= urlAPI.replaceAll("xxxxxx", fechaI);
}
String urlTemp = urlAPI.replace("p_id", "TEMP");
String urlHum = urlAPI.replace("p_id", "HUM");
String urlPres = urlAPI.replace("p_id", "PRES");
String urlCO2 = urlAPI.replace("p_id", "CO2");
URL urlTemperatura = new URL(urlTemp);
URL urlHumitat = new URL(urlHum);
URL urlPressio = new URL(urlPres);
URL urlConcentracion = new URL(urlCO2);

URLConnection temp = urlTemperatura.openConnection();
temp.connect();

URLConnection hum = urlHumitat.openConnection();
hum.connect();

URLConnection pres = urlPressio.openConnection();
pres.connect();

URLConnection co2 = urlConcentracion.openConnection();
co2.connect();
#>
```

**Figura 29.** Codi Java pel reemplaçament dels paràmetres

Una vegada tenim els quatre enllaços i una connexió establerta, llegirem les dades de l'API contingudes al seu interior i ens quedarem amb les cinc últimes de cadascuna, aquestes seran col·locades en una taula.

Per acabar amb la creació de mòduls, s'ha desenvolupat el mòdul acordió. Com indica el seu propi nom aquest constarà de diversos acordions. Veiem la seva estructura.

### Inspect TFG\_Module\_CD: TFG\_ModuleAccordion

---

**\*Name:** TFG\_ModuleAccordion

**Description:**

**Status:** Created

**ID:** 1621239878207

**Parent Definitions:** (none)

**Attribute Names:** TFG\_ModuleTitulo (S)  
TFG\_ModuleDescripcion (S)  
TFG\_ModuleAccordion (M)

### Figura 30. Mòdul Accordion

Aquest consta de tres atributs: títol, descripció i acordió. L'últim és un *Asset* de tipus contingut i subtipus *Content Accordion*.

### TFG\_Module\_A: TFG\_ModuleAccordion

---

**\*Name:** TFG\_ModuleAccordion

**Description:** Acordeon(es)

**Status:** Edited

**ID:** 1621239878198

**Site:** Site exclusivo TFG Esther

**Attribute Type:** asset

**Asset Type:** TFG\_Content\_P

**Asset Subtypes:** TFG\_ContentAccordion

### Figura 31. Atribut del contingut acordió

Passem a analitzar aquest subtipus de contingut.

### Inspect TFG\_Content\_CD: TFG\_ContentAccordion

**\*Name:** TFG\_ContentAccordion

**Description:**

**Status:** Edited

**ID:** 1621239878215

**Parent Definitions:** (none)

**Attribute Names:** TFG\_ContentTitle (S)  
TFG\_ContentAccordionRow (M)

**Figura 32.** Atributs del contingut acordió

Cada acordió podrà tindre el seu títol i diferents files. El títol és de tipus text, però les files també són d'un altre subtipus de contingut, *Content Accordion Row*.

### TFG\_Content\_A: TFG\_ContentAccordionRow

**\*Name:** TFG\_ContentAccordionRow

**Description:** Filas del acordeón

**Status:** Edited

**ID:** 1621239878333

**Site:** Site exclusivo TFG Esther

**Attribute Type:** asset

**Asset Type:** TFG\_Content\_P

**Asset Subtypes:** TFG\_ContentAccordionRow

**Figura 33.** Atribut de tipus ContentAccordionRow

### Inspect TFG\_Content\_CD: TFG\_ContentAccordionRow

**\*Name:** TFG\_ContentAccordionRow

**Description:** Fila del acordeón

**Status:** Edited

**ID:** 1621239878326

**Parent Definitions:** (none)

**Attribute Names:** TFG\_ContentTitle (S)  
TFG\_ContentDescripcion (S)  
TFG\_ContentLinkTitle (S)  
TFG\_ContentLinkHref (S)

**Figura 34.** Atributs del contingut fila del acordió

Cada fila de l'acordió té la possibilitat d'afegir un títol, subtítol, títol de l'enllaç i el propi enllaç.

Quan tenim l'estructura del mòdul ben definida, es crea el seu codi, aquesta vegada més complex per poder recuperar tots els seus continguts.

```

<section class="py-5">
  <div class="container px-5 my-5">
    <div class="text-center mb-5">
      <h1 class="fw-bold">${accordion.TFG_ModuleTitulo}</h1>
      <p class="lead fw-normal text-muted mb-0">${accordion.TFG_ModuleDescripcion}</p>
    </div>
    <div class="justify-content-center">
      <div class="col-xl-1">
        <:if test="!(empty accordion.TFG_ModuleAccordion)">
          <:forEach items="${accordion.TFG_ModuleAccordion}" var="itemAccordion">
            <:getattrributes name="accordion" assettype="TFG_Content_P" assetid="${itemAccordion.id}" attributes="TFG_ContentTitle,TFG_ContentAccordionRow"/>
            <h2 class="fw-bold mb-3">${accordion.TFG_ContentTitle}</h2>
            <div class="accordion mb-5 id="accordionExample">
              <:if test="!(empty accordion.TFG_ContentAccordionRow)">
                <:forEach items="${accordion.TFG_ContentAccordionRow}" var="itemAccordionRow" varStatus="numItemCont">
                  <:getattrributes name="accordionFila" assettype="TFG_Content_P" assetid="${itemAccordionRow.id}"
                    attributes="TFG_ContentTitle,TFG_ContentDescripcion,TFG_ContentLinkTitle,TFG_ContentLinkHref"/>
                  <:set var="urlJSON" value="${accordionFila.TFG_ContentLinkHref}"/>
                  <:choose>
                    <:when test="${numItemCont.first}">
                    <:when test="${numItemCont.last}">
                    <:otherwise>
                    </c:choose>
                </c:forEach>
              </c:if>
            </div>
          </c:forEach>
        </div>
      </div>
    </div>
  </div>
</section>

```

**Figura 35.** Codi del mòdul Accordion

Per començar dins de l'estructura, col·locarem el títol i la descripció. Després d'això, es fa un bucle per recuperar cadascun dels acordions. En tercer lloc, quan ja estem dins de cada acordió, es torna a fer un altre bucle per recuperar cadascun dels atributs de cada fila. És en aquest moment on es comprova si la fila és la primera, l'última o altres casos, ja que les classes HTML són diferents i aquestes afecten el Java Script que hem importat.

A cadascuna de les files es crida a l'element de codi esmentat anteriorment: lecturaJson. Recordem que aquest és l'encarregat de reemplaçar la data inicial i final de l'enllaç a l'API i mostrar una taula amb les últimes cinc dades recollides.

```

<div class="accordion-item">
  <h3 class="accordion-header" id="headingOne"><button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne" aria-expanded="true"
    aria-controls="collapseOne">${accordionFila.TFG_ContentTitle}</button></h3>
  <div class="accordion-collapse collapse show" id="collapseOne" aria-labelledby="headingOne" data-bs-parent="#accordionExample">
    <div class="accordion-body">
      <p>${accordionFila.TFG_ContentDescripcion}</p>
      <:render:callement elementname="tfg/lecturajson" scoped="global">
        <:render:argument name="c" value="TFG_Content_P" />
        <:render:argument name="urlAtr" value="${accordionFila.TFG_ContentLinkHref}" />
        <:render:argument name="site" value="${args.site}" />
      </render:callement>
      <div class="fst-italic text-muted">
        <a href="${urlAPI}">${accordionFila.TFG_ContentLinkTitle}</a>
      </div>
    </div>
  </div>
</div>

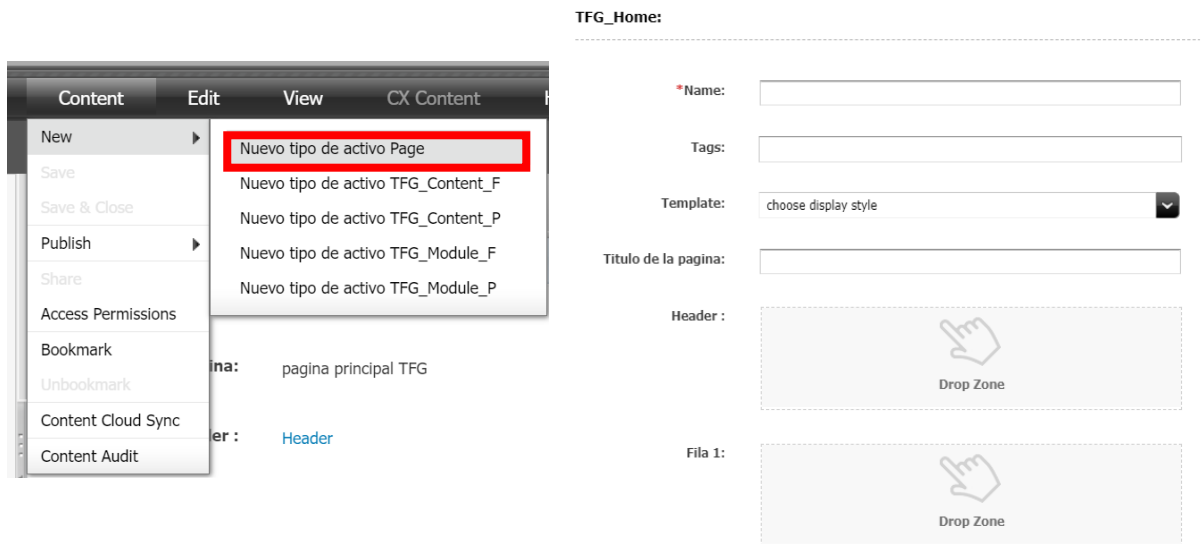
```

**Figura 36.** Crida a l'element lecturajson

#### 4.2.2. Gestió dels continguts

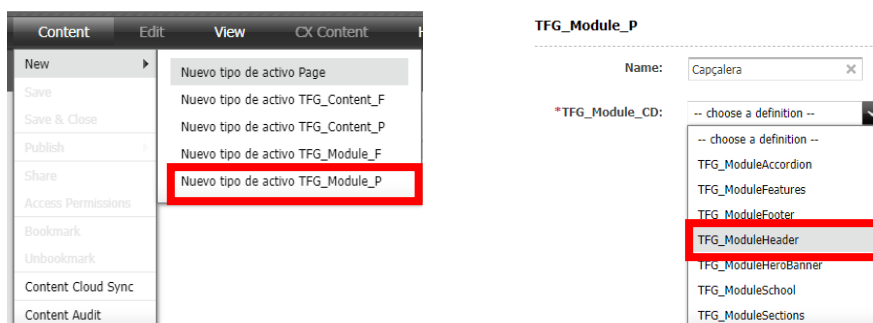
En aquest apartat es troben les indicacions per tal d'afegir contingut al lloc web sense tindre la necessitat de disposar de coneixements tècnics, tot això ho farem des de la interfície de contribució.

El primer que hem de fer és la creació d'una pàgina. A partir d'aquí ja podem començar a introduir els diferents continguts.



**Figura 37.** Creació d'una pàgina

Seguidament, crearem la capçalera i la introduïrem dins de la nostra pàgina. També podem afegir enllaços dins d'aquesta si fos necessari, aquests haurien d'anar a l'atribut enllaços de la capçalera.



**Figura 38.** Creació de la capçalera

**TFG\_ModuleHeader: Capçalera**

---


**\*Name:**

**Tags:**

**Template:**  ▼

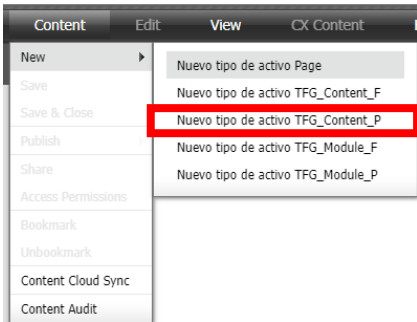
**Título del Header:**  ✕

**Enlaces del header:**

  
**Drop Zone**

**Figura 39.** Aspecte de contribució de la capçalera

Pel cas del enllaços, crearem un contingut de tipus Content i de subtipus *ContentLink*.



**TFG\_Content\_P**

**Name:**

**\*TFG\_Content\_CD:** -- choose a definition -- ▼

- choose a definition --
- TFG\_ContentAccordion
- TFG\_ContentAccordionRow
- TFG\_ContentFeatures
- TFG\_ContentImage
- TFG\_ContentImageLink
- TFG\_ContentLink
- TFG\_ContentSection

**Figura 40.** Creació d'un contingut tipus enllaç

Afegirem la informació del títol de l'enllaç, és obligatori, i l'enllaç extern al qual anirem.

**\*Name:**  ✕

**Tags:**

**Template:**  ▼

**\* Título enlace:**

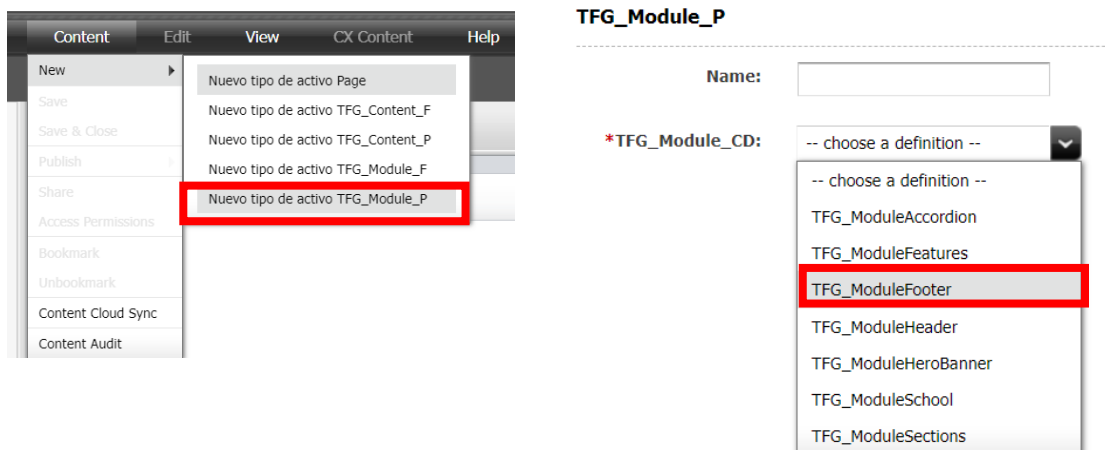
**Href enlace:**

**Figura 41.** Formulari de contribució d'un enllaç

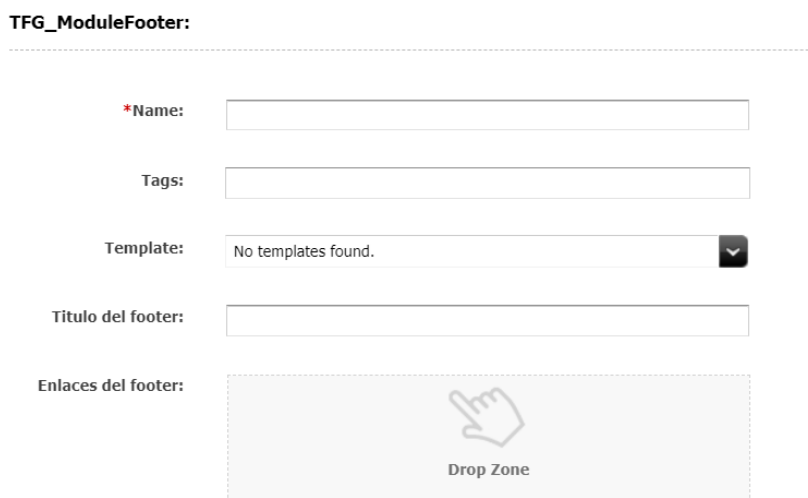
Quan tinguem l'enllaç creat només l'haurem d'arrossegat fins on el vulguem incloure, en aquest cas, a la capçalera.

Quan tinguem la capçalera acabada també l'arrossegarem fins a la nostra pàgina creada anteriorment. Cal destacar que podrem editar els continguts tantes vegades com fos necessari.

Posteriorment, contribuïrem el mòdul del peu de pàgina.



**Figura 42.** Creació del peu de pàgina



**Figura 43.** Contribució del peu de pàgina

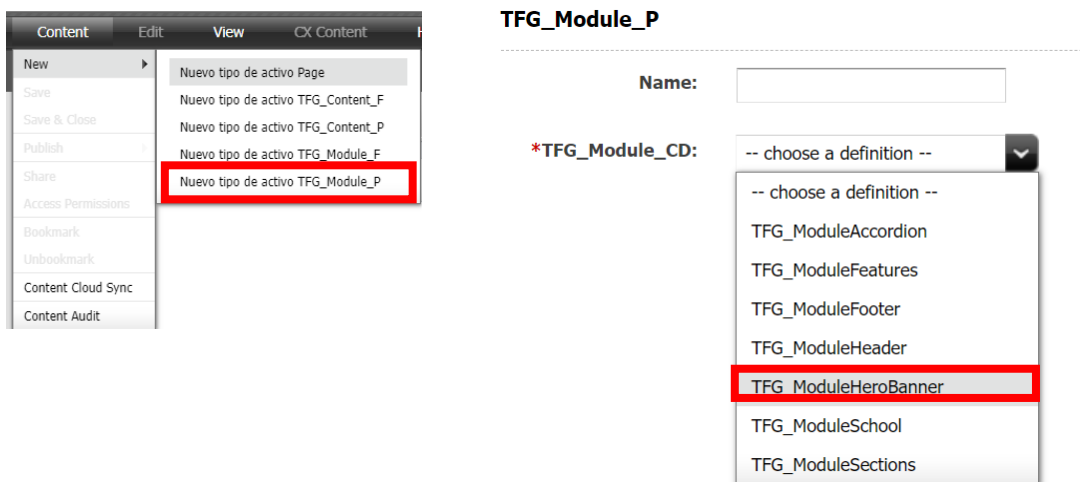
Igual que en la capçalera, inclourem els enllaços que seran de tipus *ContentLink*.

Una vegada tenim l'estructura d'una pàgina creada, capçalera i peu de pàgina, és el moment de començar a introduir els diferents continguts. Com hem vist a l'apartat anterior, tenim la possibilitat d'incloure cinc mòduls totalment diferents, cadascun d'ells els podem afegir al nostre portal web tantes vegades com vulguem amb el contingut que necessitem en cada moment. Amb aquests cinc mòduls diferenciats que han estat creats gràcies a un desenvolupament tècnic podrem construir un lloc web molt ampli.

Començarem amb un mòdul introductor, aquest l'utilitzarem dins del portal web per introduir el projecte a la pàgina principal i per presentar cadascuna de les escoles de les quals donarem més detall. És important recalcar que ho podríem utilitzar en qualsevol lloc del portal que ho necessitem.

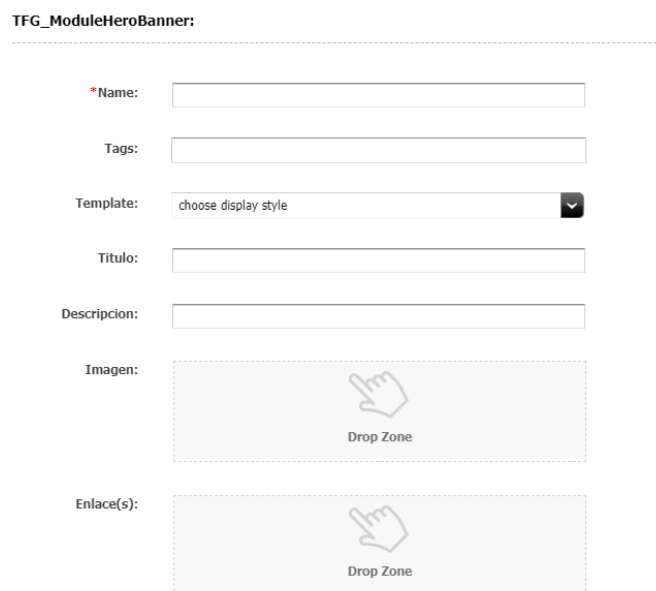
Es tracta del mòdul anomenat *Hero Banner* i té la possibilitat d'afegir un títol, una descripció, diferents enllaços en forma de botó i una imatge. Els passos que hem de seguir per crear-lo s'expliquen a continuació.

Primerament, hem de triar el tipus, i el subtipus. Com és un mòdul, el tipus d'*Asset* és *Module* i el subtipus *Hero Banner*.



**Figura 44.** Creació del mòdul Hero Banner

Podem observar quina és la plantilla d'aquest mòdul.



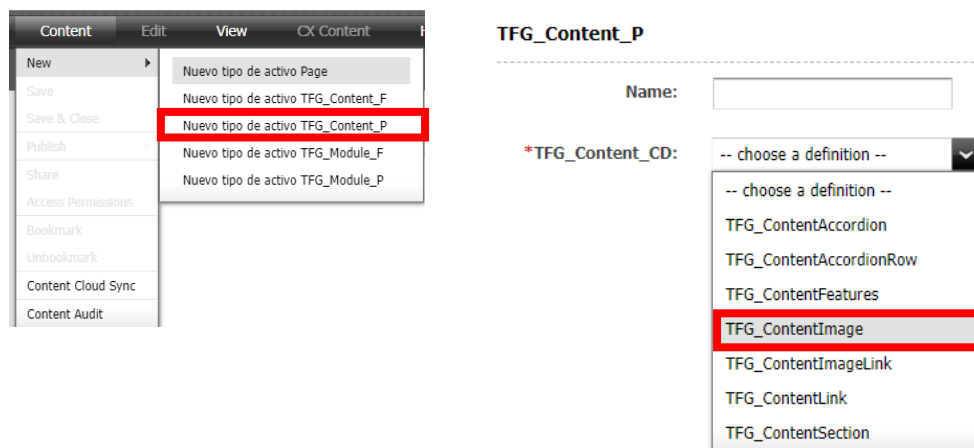
**Figura 45.** Contribució del mòdul Hero Banner

Aquesta és la plantilla que hem d'omplir cada vegada que vulguem crear aquest mòdul. És molt important seleccionar el tipus de *Template*, és un desplegable i només trobarem la plantilla del nostre mòdul, és aquesta la que hem de seleccionar.

El títol i la descripció són camps de lliure escriptura, però la imatge i els enllaços són de tipus *Asset*, per tant, els hem de crear i arrossegar-los fins aquí.

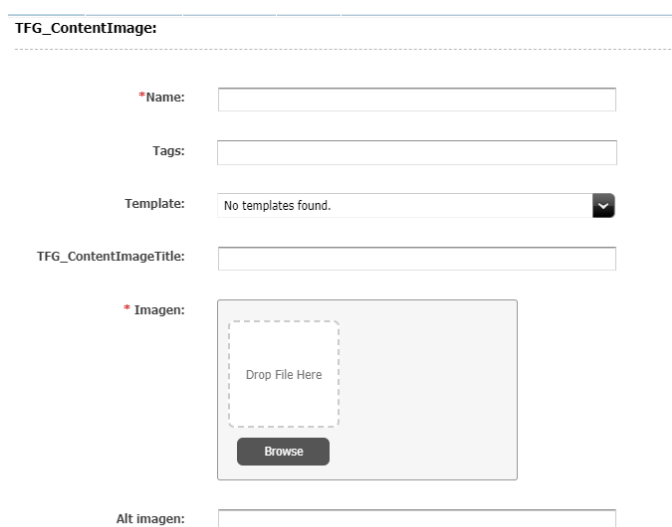
Ja hem parlat anteriorment de com crear un tipus de contingut enllaç (*Content Link*) i ara parlarem d'un contingut imatge (*Content Image*).

Com sempre, hem de triar el tipus i el subtipus, en aquest cas, *Content* i *Content Image*.



**Figura 46.** Creació del contingut imatge

Per contribuir la imatge hem d'omplir el camp títol, imatge i alt de la imatge. No és obligatori la contribució del títol ni del text alternatiu, però sí que és obligatori la contribució de la imatge. Aquesta la podem arrossegar des de l'explorador d'arxius del nostre ordinador o fent clic al botó "*Browse*" se'ns obrirà l'explorador d'arxius per a poder triar la imatge.



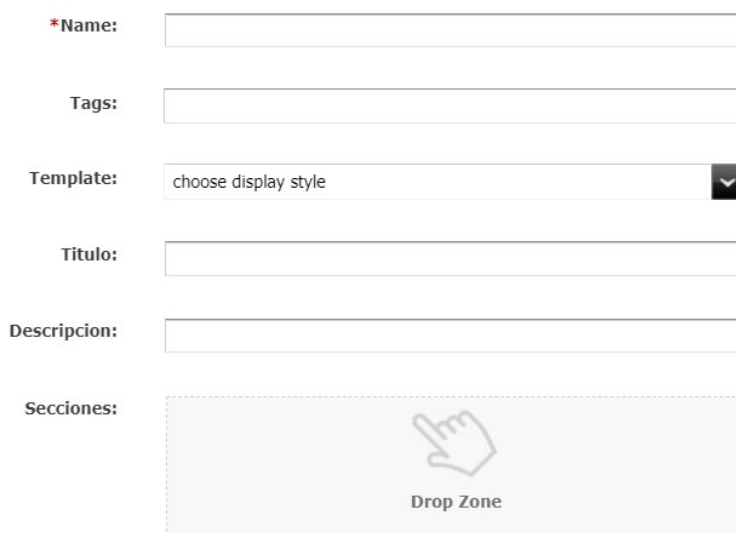
**Figura 47.** Contribució d'una imatge

Una vegada tenim el contingut imatge llest ja el podem arrossegar fins al nostre mòdul.

Després d'haver entès com es contribueix un mòdul, tots els altres seran molt semblants, però cadascun amb les seves pròpies característiques.

El segon mòdul rep el nom de *Sections* i consta d'un títol i descripció i diferents seccions. Com sempre hem de triar el tipus i subtipus, sent de subtipus *Sections*.

#### TFG\_ModuleSections:



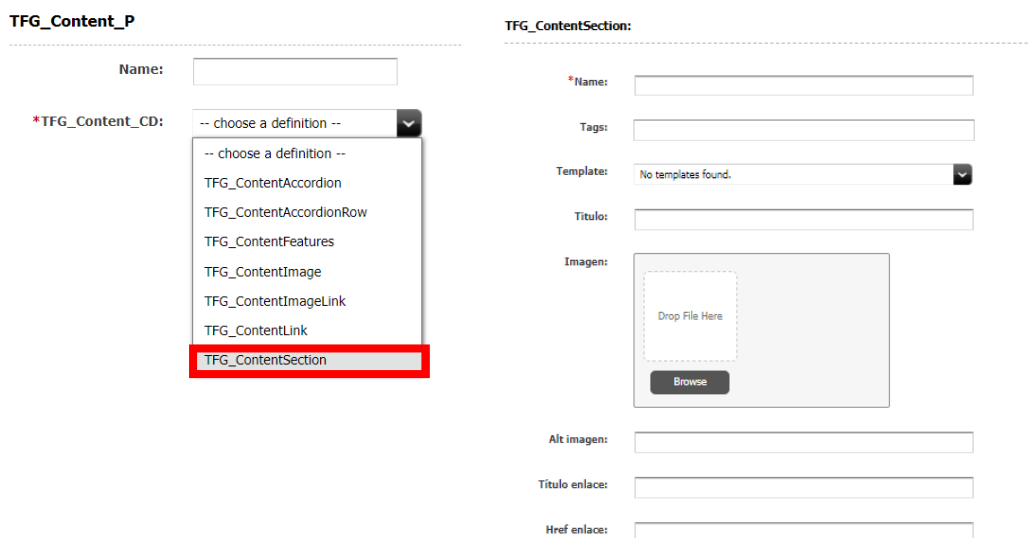
The form for TFG\_ModuleSections contains the following fields:

- \*Name:** A text input field.
- Tags:** A text input field.
- Template:** A dropdown menu with the text "choose display style" and a downward arrow.
- Título:** A text input field.
- Descripcion:** A text input field.
- Secciones:** A dashed rectangular area containing a hand icon and the text "Drop Zone".

**Figura 48.** Contribució del mòdul Sections

Veiem com el títol i la descripció són camps per introduir text, però les seccions són de tipus *Asset*. Per tant, hem de crear aquests continguts i arrossegar-los fins aquí.

Per crear el contingut de les *Sections* seleccionarem el seu subtipus i afegirem tota la informació que vulguem contribuir.



The image shows two forms side-by-side:

**TFG\_Content\_P**

- Name:** A text input field.
- \*TFG\_Content\_CD:** A dropdown menu with a list of options: "-- choose a definition --", "TFG\_ContentAccordion", "TFG\_ContentAccordionRow", "TFG\_ContentFeatures", "TFG\_ContentImage", "TFG\_ContentImageLink", "TFG\_ContentLink", and "TFG\_ContentSection". The "TFG\_ContentSection" option is highlighted with a red border.

**TFG\_ContentSection:**

- \*Name:** A text input field.
- Tags:** A text input field.
- Template:** A dropdown menu with the text "No templates found." and a downward arrow.
- Título:** A text input field.
- Imagen:** A file upload area with a dashed box containing "Drop File Here" and a "Browse" button.
- Alt imagen:** A text input field.
- Título enlace:** A text input field.
- Href enlace:** A text input field.

**Figura 49.** Creació i contribució d'un contingut Section

És important recordar que podem afegir tantes seccions com fos necessari al nostre mòdul *Sections*.

El tercer mòdul creat s'anomena *Features* i la seva contribució és molt semblant a la del mòdul *Sections*.

Primer de tot es crea el mòdul on es pot incloure un títol i diferents característiques.

The image shows two forms side-by-side. The left form is titled 'TFG\_Module\_P' and has a 'Name:' text input field. Below it is a dropdown menu for '\*TFG\_Module\_CD:' with the text '-- choose a definition --'. The dropdown menu is open, showing a list of options: '-- choose a definition --', 'TFG\_ModuleAccordion', 'TFG\_ModuleFeatures' (highlighted with a red box), 'TFG\_ModuleFooter', 'TFG\_ModuleHeader', 'TFG\_ModuleHeroBanner', 'TFG\_ModuleSchool', and 'TFG\_ModuleSections'. The right form is titled 'TFG\_ModuleFeatures:' and has a '\*Name:' text input field, a 'Tags:' text input field, a 'Template:' dropdown menu with 'choose display style' selected, a 'Titulo:' text input field, and a 'Features:' area which is a dashed box containing a hand icon and the text 'Drop Zone'.

**Figura 50.** Creació i contribució del mòdul Features

Per crear cadascuna de les característiques, necessitem crear el tipus de contingut *ContentFeatures* i omplir els atributs que necessitem mostrar.

The image shows two forms side-by-side. The left form is titled 'TFG\_Content\_P' and has a 'Name:' text input field. Below it is a dropdown menu for '\*TFG\_Content\_CD:' with the text '-- choose a definition --'. The dropdown menu is open, showing a list of options: '-- choose a definition --', 'TFG\_ContentAccordion', 'TFG\_ContentAccordionRow', 'TFG\_ContentFeatures' (highlighted with a red box), 'TFG\_ContentImage', 'TFG\_ContentImageLink', 'TFG\_ContentLink', and 'TFG\_ContentSection'. The right form is titled 'Features:' and has a '\*Name:' text input field, a 'Tags:' text input field, a 'Template:' dropdown menu with 'No templates found.' selected, a 'Titulo:' text input field, a 'Descripción:' text input field, an 'Imagen:' area which is a dashed box containing a 'Drop File Here' text and a 'Browse' button, an 'Alt imagen:' text input field, a 'Titulo enlace:' text input field, and a 'Href enlace:' text input field.

**Figura 51.** Creació i contribució del contingut Features

Per continuar, aprendrem a contribuir el mòdul dedicat a una escola, anomenat *School*.

The screenshot shows two panels: 'TFG\_Module\_P' and 'TFG\_ModuleSchool:'. In the 'TFG\_Module\_P' panel, the 'Name' field is empty, and the '\*TFG\_Module\_CD:' dropdown menu is open, showing a list of module definitions. 'TFG\_ModuleSchool' is highlighted with a red box. In the 'TFG\_ModuleSchool:' panel, the '\*Name:' field is empty, 'Tags:' is empty, 'Template:' is set to 'TFG\_School', 'Titulo:', 'Descripcion:', and 'Enllaç API:' fields are all empty.

**Figura 52.** Creació i contribució del mòdul School

En aquest, afegirem un títol, descripció i l'enllaç a l'API en qüestió. Recordem que pel correcte funcionament s'ha de substituir el paràmetre identificador per *p\_id*, la data inicial per xxxxx i la data final per yyyy.

Arribem a l'últim i més complex, en quant fa a la contribució, mòdul. Rep el nom d'*Accordion*. Per començar es crea l'estructura del mòdul.

The screenshot shows two panels: 'TFG\_Module\_P' and 'TFG\_ModuleAccordion:'. In the 'TFG\_Module\_P' panel, the 'Name' field is empty, and the '\*TFG\_Module\_CD:' dropdown menu is open, showing a list of module definitions. 'TFG\_ModuleAccordion' is highlighted with a red box. In the 'TFG\_ModuleAccordion:' panel, the '\*Name:' field is empty, 'Tags:' is empty, 'Template:' is set to 'choose display style', 'Titulo:', 'Descripcion:', and 'Acordeon(es):' fields are all empty. The 'Acordeon(es):' field has a dashed border and a 'Drop Zone' label with a hand icon.

**Figura 53.** Creació i contribució del mòdul acordió

Una vegada tenim creada l'estructura del mòdul, es creen els diferents continguts acordió.

**TFG\_Content\_P**

---

Name:

\*TFG\_Content\_CD: -- choose a definition --

-- choose a definition --

TFG\_ContentAccordion

TFG\_ContentAccordionRow

TFG\_ContentFeatures

TFG\_ContentImage

TFG\_ContentImageLink

TFG\_ContentLink

TFG\_ContentSection

**TFG\_ContentAccordion:**

---


\*Name:

Tags:

Template: No templates found.

Título:

Filas del acordeón: 

  
Drop Zone

**Figura 54.** Creació i contribució del contingut Acordió

Veiem com en aquest contingut s'han incloure les diferents files que formaran l'acordió.

**TFG\_Content\_P**

---

Name:

\*TFG\_Content\_CD: -- choose a definition --

-- choose a definition --

TFG\_ContentAccordion

TFG\_ContentAccordionRow

TFG\_ContentFeatures

TFG\_ContentImage

TFG\_ContentImageLink

TFG\_ContentLink

TFG\_ContentSection

**Fila del acordeón:**

---

\*Name:

Tags:

Template: No templates found.

Título:

Descripción:

Título enlace:

Href enlace:

**Figura 55.** Creació i contribució d'una fila del acordeió

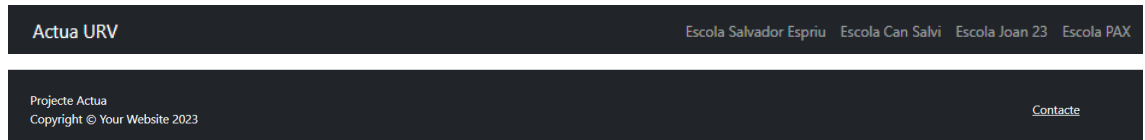
És aquí on omplirem la informació necessària per a cada fila, sempre recordant que els paràmetres de data inicial i final de l'enllaç s'han de substituir pel correcte funcionament.

### 4.2.3. Resultat final del lloc web

Una vegada el desenvolupament tècnic ha sigut realitzat i sabem com s'han de contribuir els diferents mòduls correctament ja es pot iniciar la fase de construcció del portal web.

Per començar crearem la capçalera i el peu de les diferents pàgines. En el nostre cas, crearem només un i, aquest serà contribuït a tot el lloc web.

A continuació, es mostra el resultat final de la contribució.



**Figura 56.** Capçalera i peu de pàgina del lloc web

El nostre lloc web està format per una pàgina inicial distributiva i quatre pàgines dedicades exclusivament a una escola cadascuna. En elles inclourem diferents dades atmosfèriques en relació amb l'escola.

Per començar, contribuïrem la pàgina distributiva inicial. Aquesta està formada per un mòdul Hero Banner que ens introdueix el projecte Actua, dos mòduls *Section*, el primer amb dades de la temperatura de quatre escoles i el segon amb dades referents a la pressió baromètrica de quatre escoles diferents de les anteriors. Seguidament, trobem un mòdul *Features* amb un enllaç directe de la temperatura, humitat, pressió baromètrica i concentració de CO2 de l'escola de Tivenys. Finalment, hi ha un mòdul *School* referent a l'escola Guillem Fortuny de Cambrils.

## Projecte Actua

Anàlisi Contextual dels factors de mitigació de la Transmissió de la COVID19 a l'aula

Més informació



## Les nostres escoles

A continuació, es mostren algunes de les escoles on s'ha realitzat l'estudi i les dades de les seves temperatures.



Escola La Pineda - Tivenys

[Veure les dades](#)  
[Veure les dades dels últims 7 dies](#)

Temperatura (°C)	Data
30.0	16:13:42
29.9	16:23:12
29.9	16:23:42
29.8	16:33:12
29.9	16:33:42



Escola Nou Patufet - Barcelona

[Veure les dades](#)  
[Veure les dades dels últims 7 dies](#)

Temperatura (°C)	Data
29.1	15:17:59
29.5	15:27:53
29.2	15:27:59
29.5	15:37:53
29.1	15:37:59



Escola Madorell - Molins de Rei

[Veure les dades](#)  
[Veure les dades dels últims 7 dies](#)

Temperatura (°C)	Data
28.4	15:13:31
28.4	15:23:10
28.4	15:23:31
28.4	15:33:10
28.4	15:33:31



Col·legi Claret - Valls

[Veure les dades](#)  
[Veure les dades dels últims 7 dies](#)

Temperatura (°C)	Data
27.9	14:14:10
27.6	14:23:51
27.8	14:24:11
27.7	14:33:52
27.8	14:34:11

En segon lloc, es mostren algunes de les escoles on s'ha realitzat l'estudi i les dades de les seves pressions baromètriques.



Escola Saavedra - Tarragona

[Veure les dades](#)  
[Veure les dades dels últims 7 dies](#)

Pressió baromètrica (hPa)	Data
1009.6	10:19:15
1009.8	10:29:15
1009.9	10:39:15
1010.0	10:49:15
1010.0	10:59:16



Escola La Canaleta - Vila-seca

[Veure les dades](#)  
[Veure les dades dels últims 7 dies](#)

Pressió baromètrica (hPa)	Data
1009.7	14:23:30
1009.8	14:33:31
1009.7	14:43:31
1009.6	14:53:31
1009.5	15:03:31



Escola El Miracle - Tarragona

[Veure les dades](#)  
[Veure les dades dels últims 7 dies](#)

Pressió baromètrica (hPa)	Data
1014.1	11:56:40
1014.0	12:06:40
1014.0	12:16:40
1014.1	12:26:40
1014.1	12:36:40



Escola Josep Veciana - Perafort

[Veure les dades](#)  
[Veure les dades dels últims 7 dies](#)

Pressió baromètrica (hPa)	Data
1002.2	15:45:35
1002.1	15:55:35
1002.3	16:05:35
1002.5	16:15:36
1002.7	16:25:36

## Escola de Tivenys



Temperatura

[Veure les dades](#)



Humitat

[Veure les dades](#)



Pressió baromètrica

[Veure les dades](#)



Concentració de CO2

[Veure les dades](#)

## Escola Guillem Fortuny

L'escola Guillem Fortuny es troba al municipi de Cambrils

[Veure la temperatura dels últims 7 dies](#)

[Veure la humitat dels últims 7 dies](#)

[Veure la pressió dels últims 7 dies](#)

[Veure la concentració de CO2 dels últims 7 dies](#)

	Temperatura	Humitat	Pressió	Concentració
13:38:46	27.8	44.6	1013.5	422.2
13:47:31	27.6	49.7	1013.3	421.8
13:48:46	27.8	44.6	1013.2	416.8
13:57:31	27.7	49.7	1013.3	418.0
13:58:46	27.8	44.5	1013.3	424.4

Figura 57. Pàgina inicial del projecte amb WCS

La segona pàgina conté únicament informació de l'escola Salvador Espriu de Roda de Berà. Aquesta contindrà un mòdul *Hero Banner* introductor a l'escola amb un enllaç directa a la seva pàgina web i cinc mòduls *School* amb informació de cadascuna de les classes les quals l'API ens retorna les seves dades.

## Escola de Salvador Espriu

En aquesta pàgina observarem diferents dades atmosfèriques de l'escola Salvador Espriu Roda de Berà

[Més informació de l'escola](#)



## Classe P3 A

[Veure la temperatura dels últims 7 dies](#)

[Veure la humitat dels últims 7 dies](#)

[Veure la pressió dels últims 7 dies](#)

[Veure la concentració de CO2 dels últims 7 dies](#)

	Temperatura	Humitat	Pressió	Concentració
12:01:00	28.4	47.6	1011.2	548.1
12:03:22	29.4	42.3	1011.2	549.7
12:11:00	28.4	47.6	1011.1	541.2
12:13:23	29.4	42.3	1011.0	544.7
12:21:00	28.4	47.7	1011.2	543.4

## Classe P5 B

[Veure la temperatura dels últims 7 dies](#)

[Veure la humitat dels últims 7 dies](#)

[Veure la pressió dels últims 7 dies](#)

[Veure la concentració de CO2 dels últims 7 dies](#)

	Temperatura	Humitat	Pressió	Concentració
15:10:48	28.7	47.3	1011.8	394.0
15:14:32	29.9	41.9	1011.7	397.6
15:20:48	28.7	47.3	1011.3	388.4
15:24:32	29.9	41.9	1011.2	386.5
15:30:48	28.7	47.1	1011.2	387.7

## Classe 3er B

[Veure la temperatura dels últims 7 dies](#)

[Veure la humitat dels últims 7 dies](#)

[Veure la pressió dels últims 7 dies](#)

[Veure la concentració de CO2 dels últims 7 dies](#)

	Temperatura	Humitat	Pressió	Concentració
16:51:41	28.4	46.7	1010.1	434.2
17:01:33	28.7	43.1	1010.1	438.1
17:01:41	28.4	46.5	1010.4	435.3
17:11:34	28.7	43.3	1010.4	438.0
17:11:41	28.4	46.6	1010.4	434.8

## Classe 4t A

[Veure la temperatura dels últims 7 dies](#)

[Veure la humitat dels últims 7 dies](#)

[Veure la pressió dels últims 7 dies](#)

[Veure la concentració de CO2 dels últims 7 dies](#)

	Temperatura	Humitat	Pressió	Concentració
17:11:02	28.5	47.2	1010.7	404.1
17:20:52	28.8	45.8	1010.8	408.0
17:21:02	28.5	47.2	1010.7	405.6
17:30:52	28.8	45.8	1010.7	409.3
17:31:03	28.5	47.2	1010.6	403.8

## Classe 6è A

[Veure la temperatura dels últims 7 dies](#)

[Veure la humitat dels últims 7 dies](#)

[Veure la pressió dels últims 7 dies](#)

[Veure la concentració de CO2 dels últims 7 dies](#)

	Temperatura	Humitat	Pressió	Concentració
15:44:07	28.7	43.5	1010.0	449.4
15:51:39	28.5	46.4	1009.8	446.5
15:54:07	28.7	43.5	1009.8	445.2
16:01:39	28.5	46.5	1009.8	444.4
16:04:07	28.7	43.6	1009.8	443.9

Figura 58. Pàgina de l'escola Salvador Espriu del projecte amb WCS

En tercer lloc, trobem una pàgina que fa referència a l'escola Can Salvi situada a Barcelona. Aquesta conté el seu mòdul introductorïer pertinent, un mòdul acordió que inclou tres acordions diferents amb dues files cadascun i per últim, un mòdul *Features* amb sis característiques.

## Escola Can Salvi

En aquesta pàgina observarem diferents dades atmosfèriques de l'escola Can Salvi de Barcelona

Més informació de l'escola



### Concentració de partícules PM1

Classe P5 A	▼
Classe 2n B	▼

### Concentració de partícules PM2.5

Classe P5 A	▼
Classe 2n B	▼

### Concentració de partícules PM10

Classe P5 A	▼
Classe 2n B	▼

### Escola Can Salvi



Temperatura de la classe P5 A

[Dades de P5 A](#)



Temperatura de la classe 2n B

[Dades de 2n B](#)



Humitat de la classe P5 A

[Dades de P5 A](#)



Humitat de la classe 2n B

[Dades de 2n B](#)



Pressió baromètrica de la classe P5 A

[Dades de P5 A](#)



Pressió baromètrica de la classe 2n B

[Dades de 2n B](#)

**Figura 59.** Pàgina de l'escola Salvador Espriu del projecte amb WCS

Posteriorment, es mostren els diferents acordions amb les files obertes, d'aquesta manera es pot veure el seu contingut.

### Concentració de partícules PM1

Classe P5 A	
<a href="#">Veure les dades dels últims 7 dies</a>	
Concentració de partícules PM1 (ug/m3)	Data
8.0	13:38:26
7.3	13:49:08
9.1	13:59:50
8.8	14:10:31
8.5	14:21:13
Classe 2n B	

**Figura 60.** Acordió de concentració de partícules PM1 amb el contingut de la classe P5 A

### Concentració de partícules PM1

Classe P5 A	
Classe 2n B	

Classe 2n B	
<a href="#">Veure les dades dels últims 7 dies</a>	
Concentració de partícules PM1 (ug/m3)	Data
5.7	19:16:13
4.0	19:26:55
6.1	19:37:36
6.0	19:48:18
5.2	19:59:00

**Figura 61.** Acordió de concentració de partícules PM1 amb el contingut de la classe 2n B

## Concentració de partícules PM2.5

Classe P5 A	
<a href="#">Veure les dades dels últims 7 dies</a>	
Concentració de partícules PM2.5 (ug/m3)	Data
10.5	13:38:26
10.1	13:49:08
12.0	13:59:50
11.6	14:10:31
11.2	14:21:13
Classe 2n B	

**Figura 62.** Acordió de concentració de partícules PM2.5 amb el contingut de la classe P5 A

## Concentració de partícules PM2.5

Classe P5 A	
Classe 2n B	

Classe 2n B	
<a href="#">Veure les dades dels últims 7 dies</a>	
Concentració de partícules PM2.5 (ug/m3)	Data
8.6	19:16:13
5.1	19:26:55
8.5	19:37:36
8.0	19:48:18
6.7	19:59:00

**Figura 63.** Acordió de concentració de partícules PM2.5 amb el contingut de la classe 2n B

## Concentració de partícules PM10

Classe P5 A	
<a href="#">Veure les dades dels últims 7 dies</a>	
Concentració de partícules PM10 (ug/m3)	Data
11.7	13:38:26
10.3	13:49:08
12.8	13:59:50
12.5	14:10:31
11.2	14:21:13
Classe 2n B	

**Figura 64.** Acordió de concentració de partícules PM10 amb el contingut de la classe P5 A

## Concentració de partícules PM10

Classe P5 A	▼
Classe 2n B	▲
<a href="#">Veure les dades dels últims 7 dies</a>	
Concentració de partícules PM10 (ug/m3)	Data
8.6	19:16:13
6.0	19:26:55
8.5	19:37:36
8.0	19:48:18
6.7	19:59:00

**Figura 65.** Acordió de concentració de partícules PM10 amb el contingut de la classe 2n B

En quart lloc, trobem l'escola Joan 23. Aquesta pàgina inclou un mòdul introductori, dos mòduls *School*, cadascun amb la informació d'una classe de l'escola i dos mòduls *Sections*, amb les dades del soroll, moviment, lluminositat i radiació UV de les classes 2n B i 4t C.

## Escola Jaon 23

En aquesta pàgina observarem diferents dades atmosfèriques de l'escola Joan 23

Més informació de l'escola



### Classe 2n B

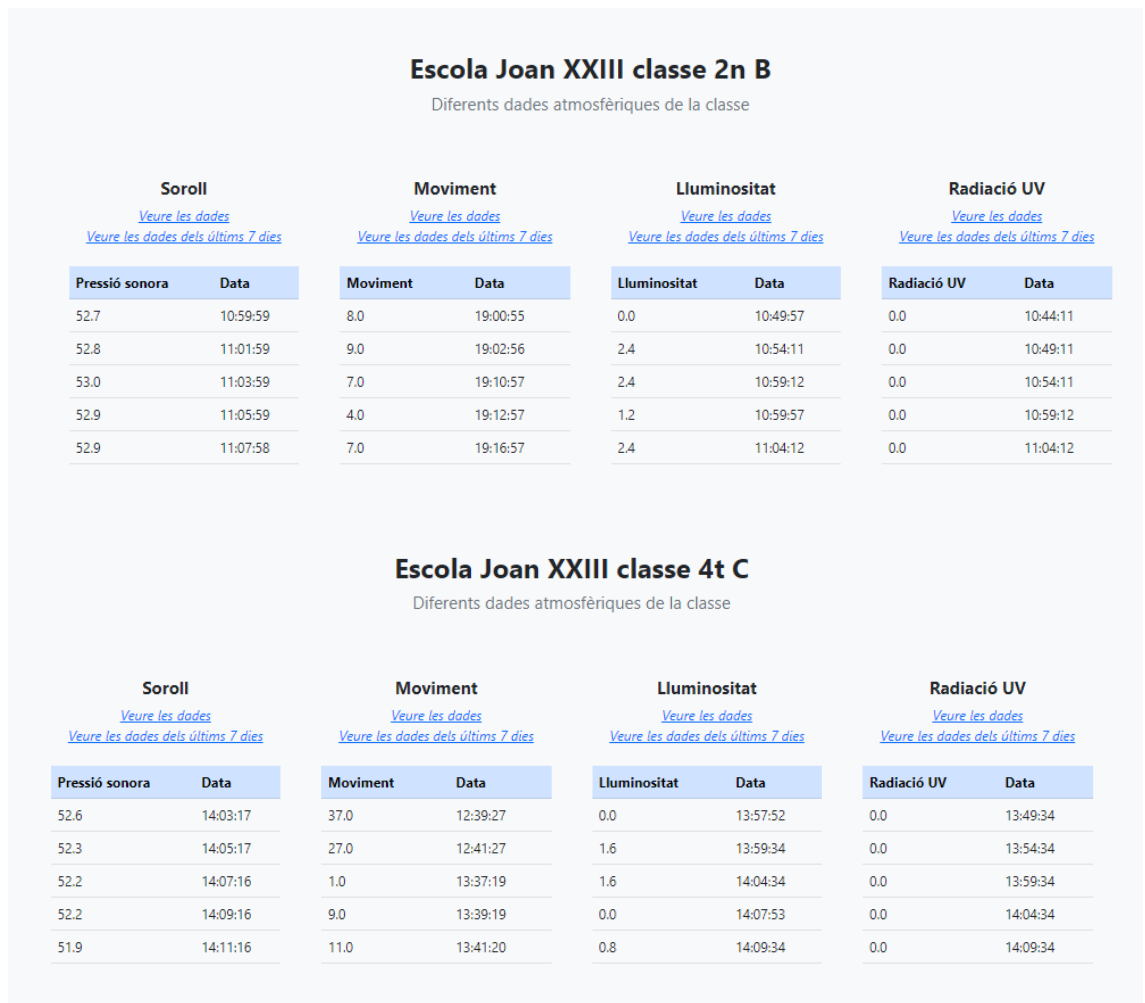
[Veure la temperatura dels últims 7 dies](#)  
[Veure la humitat dels últims 7 dies](#)  
[Veure la pressió dels últims 7 dies](#)  
[Veure la concentració de CO2 dels últims 7 dies](#)

	Temperatura	Humitat	Pressio	Concentracio
10:39:56	27.1	46.4	1009.1	604.2
10:49:48	27.1	49.6	1009.2	613.4
10:49:57	27.1	46.4	1009.4	606.5
10:59:48	27.1	49.7	1009.5	611.1
10:59:57	27.1	46.4	1009.5	611.0

### Classe 4t C

[Veure la temperatura dels últims 7 dies](#)  
[Veure la humitat dels últims 7 dies](#)  
[Veure la pressió dels últims 7 dies](#)  
[Veure la concentració de CO2 dels últims 7 dies](#)

	Temperatura	Humitat	Pressio	Concentracio
13:38:46	27.8	44.6	1013.5	422.2
13:47:31	27.6	49.7	1013.3	421.8
13:48:46	27.8	44.6	1013.2	416.8
13:57:31	27.7	49.7	1013.3	418.0
13:58:46	27.8	44.5	1013.3	424.4



**Figura 66.** Pàgina de l'escola Joan 23 del projecte amb WCS

Per acabar, l'última pàgina que inclou el lloc web tracta sobre l'escola PAX. Aquí hi trobarem una breu introducció, amb un enllaç directe a la seva pàgina web, i un mòdul acordió que conté tres continguts acordions, amb les dades de la temperatura, pressió sonora i lluminositat, cadascun amb tres files que fan referència a les classes de P5A, 1r A i 5è A.

## Escola Pax

En aquesta pàgina observarem diferents dades atmosfèriques de l'escola PAX

Més informació de l'escola



Es poden observar diferents dades atmosfèriques de diferents classes de l'escola

### Dades respecte a la temperatura

Classe P5 A	▼
Classe 1r A	▼
Classe 5è A	▼

### Dades respecte a la pressió sonora

Classe P5 A	▼
Classe 1r A	▼
Classe 5è A	▼

### Dades respecte a la lluminositat

Classe P5 A	▼
Classe 1r A	▼
Classe 5è A	▼

**Figura 67.** Pàgina de l'escola PAX del projecte amb WCS

Tot seguit, es mostren els diferents acords amb les files desplegadas, d'aquesta manera es pot observar el seu contingut.

### Dades respecte a la temperatura

Classe P5 A	▼
<a href="#">Veure les dades dels últims 7 dies</a>	
Temperatura (°C)	Data
27.5	14:00:28
27.4	14:09:09
27.5	14:10:28
27.4	14:19:10
27.5	14:20:28
Classe 1r A	▼
Classe 5è A	▼

**Figura 68.** Acordió amb dades respecte la temperatura de la classe P5 B

### Dades respecte a la temperatura

Classe P5 A	▼
Classe 1r A	▲
<a href="#">Veure les dades dels últims 7 dies</a>	
Temperatura (°C)	Data
28.7	14:48:52
28.7	14:50:08
28.7	14:58:53
28.7	15:00:08
28.7	15:08:53
Classe 5è A	▼

**Figura 69.**Acordió amb dades respecte la temperatura de la classe 1r A

### Dades respecte a la temperatura

Classe P5 A	▼
Classe 1r A	▼
Classe 5è A	▼
<a href="#">Veure les dades dels últims 7 dies</a>	
Temperatura (°C)	Data
29.6	15:00:27
28.5	15:09:04
29.6	15:10:27
28.5	15:19:04
29.6	15:20:27

**Figura 70.** Acordió amb dades respecte la temperatura de la classe 5è A

## Dades respecte a la pressió sonora

Classe P5 A	^
<a href="#">Veure les dades dels últims 7 dies</a>	
Pressió sonora (dB)	Data
51.2	14:15:18
51.0	14:17:18
51.1	14:19:18
51.1	14:21:19
51.0	14:23:19
Classe 1r A	v
Classe 5è A	v

**Figura 71.** Acordió amb dades respecte la pressió sonora de la classe P5 B

## Dades respecte a la pressió sonora

Classe P5 A	v
Classe 1r A	v
<a href="#">Veure les dades dels últims 7 dies</a>	
Pressió sonora (dB)	Data
50.3	15:00:25
49.8	15:02:25
49.8	15:04:25
48.9	15:06:25
49.8	15:08:25
Classe 5è A	v

**Figura 72.** Acordió amb dades respecte la pressió sonora de la classe 1r A

## Dades respecte a la pressió sonora

Classe P5 A	v
Classe 1r A	v
Classe 5è A	v
<a href="#">Veure les dades dels últims 7 dies</a>	
Pressió sonora (dB)	Data
50.6	15:13:44
50.5	15:15:46
49.6	15:17:46
50.5	15:19:47
50.6	15:21:47

**Figura 73.** Acordió amb dades respecte la pressió sonora de la classe 5è A

## Dades respecte a la lluminositat

Classe P5 A	▼
<a href="#">Veure les dades dels últims 7 dies</a>	
Lluminositat (lux)	Data
11.2	14:10:00
4.1	14:10:28
9.6	14:15:00
9.6	14:20:00
3.6	14:20:28
Classe 1r A	▼
Classe 5è A	▼

**Figura 74.** Acordió amb dades respecte la lluminositat de la classe P5 B

## Dades respecte a la lluminositat

Classe P5 A	▼
Classe 1r A	▼
<a href="#">Veure les dades dels últims 7 dies</a>	
Lluminositat (lux)	Data
69.6	14:50:13
71.2	14:55:13
13.3	15:00:08
72.8	15:00:13
73.6	15:05:13
Classe 5è A	▼

**Figura 75.** Acordió amb dades respecte la lluminositat de la classe 1r A

## Dades respecte a la lluminositat

Classe P5 A	▼
Classe 1r A	▼
Classe 5è A	▼
<a href="#">Veure les dades dels últims 7 dies</a>	
Lluminositat (lux)	Data
12.0	15:10:06
1.9	15:10:27
12.8	15:15:07
13.6	15:20:07
2.2	15:20:27

**Figura 76.** Acordió amb dades respecte la lluminositat de la classe 5è A

## 4.3. Wordpress

Com ja sabem, Wordpress és un sistema de gestió de continguts (CMS), on es pot crear i gestionar un lloc web de manera senzilla i eficient. Inclou una interfície amb eines amigables i intuïtives. Conté una àmplia varietat de temes i extensions disponibles per a poder un lloc web segons les seves necessitats. Seguidament, s'explicarà com s'ha realitzat la instal·lació en una màquina macOS, el funcionament de Wordpress i finalment els resultats obtinguts.

### 4.3.1. Instal·lació

Per poder dur a terme aquest projecte, vaig haver de fer la instal·lació completa de Wordpress al meu ordinador, Apple Mac Book Pro M1.

En primer moment vaig instal·lar UTM<sup>7</sup>, una aplicació de virtualització de codi obert per a macOS. UTM permet executar sistemes operatius i màquines virtuals en Mac, i això permet emular diferents arquitectures com ARM o x86.

Una vegada tenim l'aplicació de UTM, hem d'instal·lar, per a després poder executar, un sistema operatiu. Des de la mateixa llibreria d'UTM podem descarregar diversos d'ells i vaig triar el sistema operatiu Debian amb la seva onzena versió. Un cop estava llest, era el moment de començar tot el llarg procés per arribar al nostre objectiu, Wordpress. Per començar s'havia de fer la instal·lació de LAMP, que a continuació explicarem de manera més detallada que és i quines són les seves funcions.

LAMP (Linux, Apache, MySQL, PHP) és una pila de software generalment utilitzada per crear i executar aplicacions webs. Cadascun dels seus components, fa una funció específica i, en conjunt, ofereixen un entorn complet per al desenvolupament d'aplicacions webs dinàmiques. Per tal de fer la seva correcta instal·lació s'han de seguir una sèrie de passes i totes d'elles les farem des del terminal.

Com a punt inicial, configurarem el servidor Apache i ho farem introduint aquesta línia al terminal, "*sudo apt install apache2*". Una vegada la instal·lació sigui completa podrem continuar amb el següent pas que és instal·lar PHP, necessari perquè Wordpress es comuniqui amb la base de dades MySQL i es mostri el seu contingut dinàmic. Això ho farem amb la comanda "*sudo apt install php libapache2-mod-php php-mysql php-curl php-gd php-xml php-mbstring php-xmlrpc php-zip php-soap php-intl -y*", que serà l'encarregada de ficar en funcionament PHP i totes les seves extensions.

Per a continuar, instal·larem MySQL i la seva base de dades amb la següent comanda: "*apt install mysql-server -y*". Un cop creada, haurem d'establir un usuari i la seva corresponent contrasenya. Després, crearem la base de dades MySQL de Wordpress. Una vegada ho tenim, arribem a l'últim pas, crear un directori en la següent ruta "*/var/www/wordpress*" i un arxiu "*WordPress.conf*". Finalment, amb aquesta comanda "*sudo apt install wget -y*" descarregarem els arxius de WordPress i ja ho tindrem tot llest per a començar a treballar.

---

<sup>7</sup> UTM: Universal Turing Machine

### 4.3.2. *Funcionament*

WordPress és un CMS que ofereix una gran flexibilitat i diversitat de temes per a personalitzar el disseny del teu lloc web. Per tant, hi ha la possibilitat de triar entre una àmplia gamma de temes disponibles, que s'adapten a diferents estils i necessitats.

No obstant això, és possible que el tema específic que hi tens al cap no estigui creat per a WordPress. En aquests casos, existeix la possibilitat de personalitzar un tema existent mitjançant codi, però aquest procés pot ser molt complex i requerir coneixements tècnics especialitzats que no entren a l'abast d'aquest treball fi de grau.

Després de considerar totes aquestes opcions, he decidit un tema disponible per a WordPress gratuït que s'ajusta bastant al que tenia en ment. Encara que aquest tema gratuït ofereix certa personalització, és important destacar que hi existeixen moltíssimes limitacions en les seves funcionalitats i capacitat de personalització. Hi ha moltes coses de les quals tenia al cap que no es poden realitzar o que resulten molt difícil fent ús de diversos pluggins.

Wordpress també ofereix opcions de pagament en els seus temes, que aquests proporcionen més funcionalitats i possibilitats de personalització avançades.

Per dur a terme aquest projecte, he triat un tema anomenat Colibrí on he pogut desenvolupar el que tenia en ment, però amb les seves limitacions d'estils i funcionalitats. És molt important tenir en compte els recursos, les habilitats tècniques i els objectius del teu lloc web abans de prendre una decisió sobre el tema de WordPress a utilitzar.

### 4.3.3. *Resultat final del lloc web*

Una vegada ja hem decidit quin tema de Wordpress utilitzarem, arriba el moment de crear les diferents pàgines que necessitem.

Aquest lloc web està format per una pàgina inicial i quatre pàgines més que fan referència a quatre escoles. Aquestes contenen una capçalera i un peu de pàgina, que serà igual per les diferents pàgines que podrem tindre al portal.

### **Figura 77.** Capçalera i peu de pàgina del lloc web

En el tema triat a Wordpress, Colibrí, inclou el format de la capçalera i el peu però aquest no és personalitzable de manera gratuïta. L'únic que podem modificar són els enllaços de la capçalera, ens permet afegir tants com vulguem.

Comencem amb el disseny de la pàgina principal del lloc web, també anomenada *Home*, en aquesta trobem una breu introducció al projecte Actua, amb un títol, subtítol, [botó que ens dirigeix fins al lloc web del projecte](#) i una imatge descriptiva. Posteriorment, hi ha una secció amb algunes de les nostres escoles analitzades, les quatre primeres contenen dades de les seves temperatures i les quatre restants ens

proporcionen dades de les seves pressions baromètriques. Després d'aquest bloc, trobem les dades de la temperatura, humitat, pressió baromètrica i concentració de CO2 de l'escola de Tivenys. Per acabar, apareix un bloc amb dades de l'escola Guillem Fortuny de Cambrils, on hi ha quatre enllaços directes per veure diverses dades atmosfèriques durant una setmana completa.

## Projecte Actua

Anàlisi Contextual dels factors de mitigació de la Transformació de la COVID-19 a l'aula

MÉS INFORMACIÓ



### Les nostres escoles

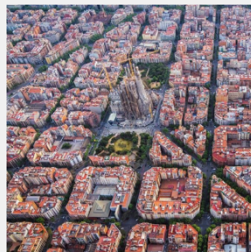
A continuació, es mostren algunes de les escoles on s'ha realitzat l'estudi i les dades de les seves temperatures.



Escola de Tivenys

[Veure les dades](#)

[Veure les dades dels últims 7 dies](#)



Escola de Barcelona

[Veure les dades](#)

[Veure les dades dels últims 7 dies](#)



Escola de Molins de Rei

[Veure les dades](#)

[Veure les dades dels últims 7 dies](#)



Escola de Valls

[Veure les dades](#)

[Veure les dades dels últims 7 dies](#)

En segon lloc, es mostren algunes de les escoles on s'ha realitzat l'estudi i les dades de les seves pressions baromètriques.



Escola Saavedra - Tarragona

[Veure les dades](#)

[Veure les dades dels últims 7 dies](#)



Escola La Canaleta - Vilaseca

[Veure les dades](#)

[Veure les dades dels últims 7 dies](#)



Escola el Miracle - Tarragona

[Veure les dades](#)

[Veure les dades dels últims 7 dies](#)



Escola Josep Veciana - Perafort

[Veure les dades](#)

[Veure les dades dels últims 7 dies](#)

## Escola de Tivenys



Temperatura

[Veure les dades](#)



Humitat

[Veure les dades](#)



Pressió baromètrica

[Veure les dades](#)



Concentració de CO2

[Veure les dades](#)

## Escola Guillem Fortuny

L'escola Guillem Fortuny es troba al municipi de Cambrils

[Veure la temperatura de 7 dies](#)

[Veure la humitat de 7 dies](#)

[Veure la pressió de 7 dies](#)

[Veure la concentració de CO2 de 7 dies](#)



**Figura 78.** Pàgina inicial del projecte amb WordPress

Seguidament, es mostren les diferents pàgines que fan referència a escoles concretes.

La primera pàgina que trobem a la capçalera del nostre lloc web fa referència a l'escola Salvador Espriu que es troba a Roda de Berà. En ella trobem un primer bloc introductori de l'escola, amb un enllaç que ens porta fins al seu lloc web.

En aquesta classe s'ha realitzat l'estudi en dues classes diferents, P3 A i P5 B, i tenim disponibles de cadascuna de les escoles un enllaç directe amb les diferents dades atmosfèriques que s'ha realitzat l'estudi.

## Escola Salvador Espriu

En aquesta pàgina observarem diferents dades atmosfèriques de l'escola Salvador Espriu Roda de Berà

[MÉS INFORMACIÓ DE L'ESCOLA](#)



## Classe P3 A

### TEMPERATURA

[Veure les dades](#)

### PRESSIÓ

[Veure les dades](#)

### CONCENTRACIÓ DE PARTÍCULES PM1

[Veure les dades](#)

### CONCENTRACIÓ DE PARTÍCULES PM10

[Veure les dades](#)

### QUANTITAT DE MOVIMENT

[Veure les dades](#)

### RADIACIÓ UV

[Veure les dades](#)

### HUMITAT

[Veure les dades](#)

### CONCENTRACIÓ DE CO2

[Veure les dades](#)

### CONCENTRACIÓ DE PARTÍCULES PM2.5

[Veure les dades](#)

### PRESSIÓ SONORA

[Veure les dades](#)

### LLUMINOSITAT

[Veure les dades](#)

## Classe P5 B

### TEMPERATURA

[Veure les dades](#)

### PRESSIÓ

[Veure les dades](#)

### CONCENTRACIÓ DE PARTÍCULES PM1

[Veure les dades](#)

### CONCENTRACIÓ DE PARTÍCULES PM10

[Veure les dades](#)

### QUANTITAT DE MOVIMENT

[Veure les dades](#)

### RADIACIÓ UV

[Veure les dades](#)

### HUMITAT

[Veure les dades](#)

### CONCENTRACIÓ DE CO2

[Veure les dades](#)

### CONCENTRACIÓ DE PARTÍCULES PM2.5

[Veure les dades](#)

### PRESSIÓ SONORA

[Veure les dades](#)

### LLUMINOSITAT

[Veure les dades](#)

**Figura 79.** Pàgina de l'escola Salvador Espriu del projecte amb WordPress

En segon lloc, es troba una pàgina que fa referència a l'escola Can Salvi, on tenim una introducció a l'escola i un segon bloc amb dades de temperatura, humitat i pressió baromètrica de les classes de P5 A i de 2n B.

**Escola Can Salvi**

En aquesta pàgina observarem diferents dades atmosfèriques de l'escola Can Salvi de Barcelona

MES INFORMACIÓ DE L'ESCOLA

**Escola CAN SALVI**

**Dades de diferents classes de l'escola**

 Temperatura de la classe P5 A DADES DE P5 A	 Temperatura de la classe 2n B DADES DE 2N B
 Humitat de la classe P5 A DADES DE P5 A	 Humitat de la classe 2n B DADES DE 2N B
 Pressió baromètrica de la classe P5 A DADES DE P5 A	 Pressió baromètrica de la classe 2n B DADES DE 2N B

**Figura 80.** Pàgina de l'escola Can Salvi del projecte amb WordPress

En tercer lloc, de la capçalera, tenim un enllaç a la pàgina de l'escola Joan 23. Igual que en les anteriors tenim una introducció i dos blocs en diferents colors que corresponen a les classes de 2n B i 4t C. En aquests trobem enllaços directes a l'API que ens retornarà les dades en un marge de set dies de la temperatura, humitat, pressió baromètrica i concentració de CO2 segons correspongui.

## Escola Joan 23

En aquesta pàgina observarem diferents dades atmosfèriques de l'escola Joan 23

MES INFORMACIÓ DE L'ESCOLA



## Classe 2n B

[Veure la temperatura de 7 dies](#)

[Veure la humitat de 7 dies](#)

[Veure la pressió baromètrica de 7 dies](#)

[Veure la concentració de CO2 de 7 dies](#)

## Classe 4t C

[Veure la temperatura de 7 dies](#)

[Veure la humitat de 7 dies](#)

[Veure la pressió baromètrica de 7 dies](#)

[Veure la concentració de CO2 de 7 dies](#)

**Figura 81.** Pàgina de l'escola Joan 23 del projecte amb WordPress

Per acabar trobem l'escola PAX, amb una introducció sobre l'escola i un bloc dividit en tres columnes, cadascuna d'aquestes correspon a un paràmetre atmosfèric, temperatura, pressió sonora i lluminositat. En cada columna trobem tres botons, que fan referència a les tres classes on s'ha realitzat l'estudi a l'escola en qüestió, P5 A, 1r A i 5è A.

The screenshot shows the 'Escola PAX' website. On the left, there is a header with the text 'Escola PAX' and a sub-header 'En aquesta pàgina observarem diferents dades atmosfèriques de l'escola PAX'. Below this is a blue button labeled 'MÉS INFORMACIÓ DE L'ESCOLA'. To the right is a vertical logo consisting of three stacked geometric shapes: a teal semi-circle, a purple triangle, and a yellow square with an 'X' inside. To the right of the logo is the text 'escola PAX' in a bold, sans-serif font. Below the header, the text 'Dades atmosfèriques de les classes P5 A, 1r A i 5è A' is centered. Underneath, there are three columns of buttons. The first column is titled 'TEMPERATURA' and has three buttons: 'CLASSE P5 A', 'CLASSE 1R A', and 'CLASSE 5È A'. The second column is titled 'PRESSIÓ SONORA' and has three buttons: 'CLASSE P5 A', 'CLASSE 1R A', and 'CLASSE 5È A'. The third column is titled 'LLUINOSITAT' and has three buttons: 'CLASSE P5 A', 'CLASSE 1R A', and 'CLASSE 5È A'.

**Figura 82.** Pàgina de l'escola PAX del projecte amb WordPress

## 5. Avaluació

Una vegada s'ha desenvolupat el projecte, cal fer una avaluació d'aquest, així podem esbrinar els punts forts i febles, i possibles solucions de cara a un futur.

Avaluarem els dos llocs web desenvolupats, Oracle WebCenter Sites i Wordpress.

### 5.1. Avaluació Oracle WebCenter Sites

Aquest projecte ha sigut desenvolupat en el meu ordinador, utilitzant Windows 10, per tant, totes les proves realitzades han sigut en local i comprovades per mi mateixa. Tot i això, si aquest projecte es volgués escalar i fer-ho públic, caldria provar-lo en un entorn real per assegurar-se que funcioni de manera correcta en les circumstàncies i condicions reals en les quals s'utilitzarà. Provar el portal web en un entorn real ens permetrà detectar i solucionar possibles problemes o incompatibilitats que podrien sorgir durant l'execució en un entorn de producció. Aquest procés de proves ajudarà a garantir que el desenvolupament sigui estable, rendible i capaç de complir amb els requisits i expectatives dels usuaris.

D'altra banda, Oracle WebCenter Sites ja té instaurats diversos controls de proves i, si hi ha qualsevol problema dins del codi del desenvolupador web, el portal deixa de funcionar i veurem la pantalla en blanc. Per tant, si el desenvolupament tècnic està acabat i sense cap mena d'errors podrem veure el lloc web sense cap problema.

Com ja hem vist, quan el desenvolupament tècnic ha sigut finalitzat es poden afegir els diferents continguts. S'han realitzat diferents avaluacions per portar el desenvolupament a l'extrem, i esbrinar si hi ha algun tipus d'error si no contribuïm un atribut. Totes les proves han sigut favorables i, si ens hem deixat qualsevol cosa sense afegir aquest contingut, simplement no es mostra dins del seu lloc establert en l'estructura HTML.

Per altra part, també s'han avaluat els diversos enllaços. Al desenvolupament tècnic es detalla com s'ha de contribuir l'API del projecte Actua per a que, en funció del mòdul, es modifiquin els paràmetres de data inicial i final o el paràmetre identificador de les dades atmosfèriques. Si aquest enllaç que introduïm no és correcte, no es podrà continuar amb el codi i, per tant, no es pintarà ni l'enllaç ni les taules que estiguin relacionades amb ell. El mateix passa si l'enllaç està buit.

### 5.2. Avaluació Wordpress

Igual que el desenvolupament que s'ha fet utilitzant el CMS Oracle WebCenter Sites, el desenvolupament fent servir el CMS de Wordpress s'ha realitzat en un ordinador en local, amb l'única diferència és que s'ha fet en una aplicació de virtualització de codi obert per a macOS.

No obstant això, en cas de tindre la intenció de fer públic el projecte, és imprescindible realitzar proves en un entorn real per assegurar-nos que funcioni de manera òptima en les condicions reals d'ús. És una part essencial del procés de desenvolupament de software i ens permetrà millorar la qualitat i la fiabilitat del portal web abans de la seva implementació final.

Les proves de desenvolupament realitzades al Wordpress han inclòs diverses etapes per assegurar la funcionalitat i l'aparença adequada de les diferents pàgines que inclou el portal web. S'ha comprovat que els enllaços interns i externs funcionin correctament. També s'ha avaluat la possibilitat d'afegir el títol de l'enllaç però no el

propi enllaç, i el resultat és que ens dirigeix a la mateixa pàgina on ens trobem.

D'altra banda, s'han realitzat proves de compatibilitat amb altres navegadors i resolucions, i s'ha verificat que tant en diferents navegadors com en diferents resolucions es pot visualitzar el contingut de manera correcta.

Finalment, s'han realitzat proves de carregament i velocitat, per assegurar-nos que totes les pàgines web del portal responguin de manera eficient.

## 6. Conclusió

Pel que fa als objectius que em vaig proposar en el principi d'aquest treball final de grau, m'agradaria verificar el seu assoliment.

En primer lloc, he aconseguit comprendre què és un sistema de gestió de continguts (CMS) i he adquirit un coneixement profund sobre el seu funcionament. He après que un CMS és una plataforma que permet gestionar i organitzar el contingut d'un lloc web, facilitant la seva creació i administració.

A continuació, he avaluat de manera específica els CMS d'Oracle WebCenter Sites i Wordpress. He analitzat les seves característiques, avantatges i àmbit d'aplicació, permetent-me comprendre les seves diferències i utilitats específiques en la gestió de continguts.

A més a més, he realitzat un estudi detallat de dos sistemes de gestió de continguts, explorant en profunditat les seves funcionalitats i capacitat per satisfer les necessitats dels usuaris. Aquesta anàlisi m'ha permès obtenir un coneixement ampli i crític sobre els dos CMS estudiats.

Per tal de posar en pràctica els meus coneixements, he desenvolupat un lloc web centrant-me en la part back-end utilitzant dos sistemes de gestió de continguts diferents. He après a utilitzar les funcionalitats i eines de cada plataforma per crear i gestionar el contingut de manera eficient.

Així mateix, he realitzat una comparació entre els dos sistemes de continguts, destacant els seus punts forts i febles, així com les seves diferències en l'àmbit de gestió de continguts. Aquesta comparativa m'ha permès avaluar quin CMS s'adapta millor a les necessitats d'un projecte específic.

Seguidament, he mostrat els resultats del projecte Actua, demostrant com l'aplicació dels CMS analitzats pot optimitzar la gestió de continguts d'un lloc web i proporcionar una millor experiència d'usuari.

Després d'haver realitzat aquest gran projecte i haver assolit els meus objectius principals, vull finalitzar amb les conclusions que he extret dels dos CMS utilitzats.

Tant WordPress com WCS són CMS útils, però la seva elecció depèn de les necessitats i objectius específics de cada projecte. Si es necessita una major flexibilitat i personalització, WCS pot ser una opció més adequada, tot i que pot implicar una complexitat addicional en la part de codi.

En el cas de WCS, un cop es desenvolupa i es configura, no cal modificar-lo constantment, ja que pot ser reutilitzat en diferents llocs de la pàgina web. Això facilita el manteniment a llarg termini i permet centrar-se en altres tasques importants.

WCS requereix un coneixement més tècnic per personalitzar els estils i adaptar-los al gust de l'usuari. És necessari comptar amb un desenvolupador front-end o tenir coneixements tècnics per aconseguir una aparença personalitzada.

Tot i que és possible gestionar contingut en WCS sense coneixements tècnics, pot ser una mica complicat per a usuaris no tècnics. No obstant això, amb una mica de pràctica i formació, pot ser gestionat amb èxit.

WordPress ofereix una àmplia gamma de temes gratuïts per triar. No obstant això, fer modificacions específiques en aquests temes gratuïts pot resultar complicat i requerir coneixements tècnics avançats.

És important centrar-se en el disseny triat i les possibilitats que ofereix, ja que WordPress té moltes funcionalitats disponibles. No obstant això, si es necessiten característiques molt específiques, pot ser necessari buscar solucions addicionals o optar per temes de pagament.

WordPress és relativament fàcil d'utilitzar i gestionar, fet que permet a qualsevol persona, amb o sense coneixements tècnics, administrar el seu lloc web de manera eficaç.

En conclusió, la selecció entre WordPress i WCS depèn de les necessitats individuals de cada projecte. WCS pot ser més adequat per a desenvolupadors amb coneixements tècnics, mentre que WordPress és més accessible per a usuaris no tècnics i ofereix una àmplia gamma de temes i funcionalitats.

Fent una valoració personal del meu treball fi de grau, puc afirmar que he après molt durant aquest llarg procés. L'experiència ha sigut molt enriquidora i m'ha permès ampliar els meus coneixements. Gràcies a l'aplicació pràctica dels continguts apresos, he pogut aprofundir en el tema i tindre noves perspectives de cadascun dels gestors de continguts utilitzats. Estic satisfeta amb els resultats obtinguts i segura de què tots els coneixements adquirits em seran de gran utilitat en el meu futur professional.

## 7. Bibliografia

- [1] Apache Software Foundation. <http://www.apache.org> (visitat 02/2023).
- [2] Anderson, J., Garcia, P. "Programming Web Services." Prentice Hall, 2014.
- [3] Smith, P. et al. "The problem of attacking PHP sites." The web of cryptography and security. <http://www.wcrypsec.org/artid?1234> (visitat 03/2023).
- [4] Smith, P. et al. "The problem of attacking PHP sites." The book of cryptography and security. Prentice Hall, 2015.
- [5] Oracle. "The History of Java." (visitat 04/2023).
- [6] Gosling, James. "The Java Language Environment." (visitat 04/2023).
- [7] Frangin, Pierre; Stouffer, Scott y Moskovits, Peter. "Oracle WebCenter Sites: Building Business-driven, Enterprise Portals."
- [8] García, Daniel Merchán e Janecek, Vita. "Oracle WebCenter Sites 12c: Build Next-Generation Enterprise Portals." (visitat 03/2023).
- [9] "HTML and CSS: Design and Build Websites" per Jon Duckett (visitat 03/2023).
- [10] "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" per Jennifer Niederst Robbins (visitat 03/2023).