

# **Práctica de Git**

## **Control de Versiones**

### Trabajo de desarrollo de una Web

Gestió i Desenvolupament del Software, 21/4/2021



**ANGEL GARCÍA CALLEJA - 1490917**  
**DANIEL CALVO RAMOS - 1494116**

# ÍNDICE

<b>INTRODUCCIÓN</b>	<b>2</b>
<b>DESARROLLO</b>	<b>4</b>
Pasos previos	4
Replicar l'estructura de treball base al Repositori Remot (Developer 1)	4
Aplicar el nou índex a totes les planes (Developer 2)	6
Crear, al peu, les planes de RGPD (Reglament general de protecció de dades) Política de privacitat / Política de cookies (amb Pop-up de configuració) i Avis legal (Developer 2).	6
Eliminar l'índex antic (netejar repositori). Commitejar al repositori (Developer 2).	7
Agregar al (master) i pujar la nova versió al repositori remot (Developer 2)	8
Pujar la branca (ED-B) per seguretat (Developer 2)	8
Descarregar modificacions realitzades pel (Developer 2). El client demana una nova modificació, al peu, agregant un camp de subscripció. Crear una nova branca (ED-C) ja que el client no ho té clar... Afegir canvi al repositori remot i ensenyar al client. Es publica al (master) (Developer 1)	8
El client acaba dient de treure el "suscríbete" però guardarem la branca (ED-C) amb el canvi per si es repensa... Guardem la branca al repositori remot per seguretat.	9
Descarregar i crear Mapa Web, agregar a totes les planes, i el sitemap.xml i publicar al (master) (Developer 2).	10
Descarregar última versió i agregar una pestanya d'ubicació amb l'API de MAPS (Developer 1).	11
Publicar i donar accés al professor a Bitbucket [>].	12
<b>FINALIZACIÓN</b>	<b>13</b>

# INTRODUCCIÓN

Git es un sistema de control de versiones. Un sistema de control de versiones sirve para trabajar en equipo de una manera mucho más simple y óptima cuando estamos desarrollando software. Cuando acabamos de desarrollar nuestro código, utilizamos Git para mezclar los cambios con los otros compañeros.

Los roles optados para esta práctica por cada uno de los miembros del equipo han sido: **Daniel Calvo Ramos** como **Developer1** y **Angel García Calleja** como **Developer2**.

El IDE utilizado para la realización de esta práctica ha sido SublimeText3, puesto que es bastante sencillo de utilizar y no requiere de la creación de ningún tipo de archivo en la carpeta del proyecto para funcionar.

Como repositorio remoto hemos utilizado la herramienta Bitbucket y como repositorio local una carpeta normal con los archivos html descargados del Campus Virtual.

Por último, como consola para realizar las diferentes operaciones con git, hemos utilizado Git Bash.

# DESARROLLO

## Pasos previos

Antes de empezar a desarrollar la práctica, debemos de crear un repositorio en bitbucket, que será el repositorio remoto donde commitearemos los cambios en el proyecto..

Para crear el repositorio, hemos definido como workspace: Daniel Calvo Ramos, Project name: Practica2\_\_GDS, repository name: Projecte Web y en access level hemos definido el repositorio como privado.

Una vez creado el repositorio, procedemos a crear un fichero .gitignore para que a la hora de comprobar los cambios realizados en el proyecto, ciertos archivos o carpetas sean ignorados.

Los archivos que queremos que sean ignorados son los que aparecen en la imagen de la derecha. Los añadimos al archivo y commiteamos.

```
# Fitxers i carpetes a ignorar del meu projecte:
assets/
META-INF/
WEB-INF/
robots.txt
sitemap.xml
```

Una vez hecho esto, procedemos a descargarnos el proyecto base del campus virtual. Ahora ya estamos listos para poder realizar las tareas de la práctica.

## Replicar l'estructura de treball base al Repositori Remot (Developer 1)

Primeramente, tenemos que inicializar el repositorio, para ello iremos al path donde tenemos nuestro projecte web descargado, ya que será ahí donde git realizará todas las operaciones.

Una vez allí inicializamos el repositorio (**git init**). Ahora miramos el contenido de nuestra carpeta base no commiteado (**git status**). Todos los archivos que no hayan sido commiteados aparecerán de color rojo. Cabe destacar que hemos eliminado el archivo index-2.html, ya que tendrá que ser añadido más adelante.

A continuación adicionamos la dirección del repositorio remoto (**git remote add origin <https://1494116@bitbucket.org/1494116/projecteweb.git>**). Esta dirección la encontraremos haciendo click en el botón *clone* disponible en la parte superior derecha de la pantalla, una vez estando dentro del repositorio creado anteriormente. Seguidamente nos descargamos lo que haya en el repositorio (**git pull origin master**). En este caso únicamente estará el archivo .gitignore que creamos anteriormente.

Para finalizar, subiremos todos los archivos de nuestro repositorio local al repositorio remoto.

Este proceso se basa en tres partes: Primero indicamos que archivos queremos que se suban al repositorio remoto, que en este caso como el repositorio remoto está vacío queremos subirlos todos (**git add .**). Para comprobar que los archivos se han añadido correctamente podemos realizar un git status (Foto 1).

```
dc30@LAPTOP-IE9FI5GT MINGW64 ~/workspac
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..."
    new file:   about.html
    new file:   contact.html
    new file:   index.html
    new file:   portfolio.html
    new file:   pricing-tables.html
    new file:   services.html
```

Foto 1. Git status después del primer git add .

Seguidamente commiteamos los cambios que conllevan los archivos que queremos subir (**git commit -m "Base ProyectoWeb"**). Finalmente, subimos todos los cambios a la rama master (**git push origin master**).

Después de haber realizado este proceso, si actualizamos el bitbucket, vemos que ahora contiene todos los archivos que tenemos en nuestra carpeta ProyectoWeb.

## Descargar i crear una nova branca de treball (ED-B) (Developer 2)

Para que el Developer 2 pueda tener permisos de lectura y de escritura al repositorio Bitbucket debemos de darle permisos desde el propio bitbucket.

Para ello debemos de irnos a la pestaña 'users and access' (accesible desde la opción 'Share Repository' del menú del repositorio) y una vez allí en 'users' escribimos el correo electrónico del developer y le damos permiso de admin. Para verificar que este proceso ha sido exitoso, el developer-2 debería de haber recibido un mail de invitación al repositorio.



Gravine

Foto 2 Añadir al Developer 2

Ahora, el developer-2 seleccionará el lugar base donde realizar operaciones con git. Seguidamente deberá realizar los mismos pasos que ya hizo el developer-1:

Primero, inicializar el repositorio (**git init**), seguidamente adicionar la dirección del repositorio remoto (**git remote add origin https://bitbucket.org/1494116/projecteweb.git**), y finalmente descargar los cambios del proyecto que haya en el repositorio remoto (**git pull origin master**).

A continuación, creamos una nueva rama (**git branch ED-B**) y replicamos el contenido de la rama master en la nueva rama (**git push origin ED-B**).

```
ANGEL@ANGELPC MINGW64 ~/De
$ git branch ED-B

ANGEL@ANGELPC MINGW64 ~/De
$ git branch
  ED-B
* master
```

Foto 3 Creación Branch ED-B

```
ANGEL@ANGELPC MINGW64 ~/Desktop/p2GDS (master)
$ git push origin ED-B
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create pull request for ED-B:
remote:  https://bitbucket.org/1494116/projecte-web/pu
-B&t=1
remote:
To https://bitbucket.org/1494116/projecte-web.git
* [new branch]      ED-B -> ED-B
```

Foto 4 Replicar master en ED-B

## Aplicar el nou índex a totes les planes (Developer 2)

Para empezar, debemos de movernos a la rama ED-B, ya que será allí donde se aplique el nuevo índice. Primero podemos mirar si la rama ED-B se ha creado correctamente (**git branch -r**) y también podemos mirar en qué rama nos encontramos actualmente (**git branch**). Ahora, partiendo de la rama master (es la única que hay creada ) nos movemos a la rama ED-B para trabajar allí (**git checkout ED-B**).

Seguidamente añadimos el archivo index-2.html a la carpeta de projecteWeb y sustituimos todas las referencias a index.html por index-2.html en los archivos about.html, contact.html, portfolio.html, index-2.html, prining-tables.html y services.html.

## Crear, al peu, les planes de RGPD (Reglament general de protecció de dades) Política de privacitat / Política de cookies (amb Pop-up de configuració) i Avis legal (Developer 2).

En este paso tenemos que crear 3 html nuevos para las 3 páginas que se nos pide agregar. Simplemente copiamos el esqueleto de alguna otra página html de nuestro proyecto y la editamos a nuestro gusto para que parezca otra página distinta. Seguidamente, hemos añadido en el footer de todos los archivos html una referencia hacia las páginas requeridas como se puede apreciar en la foto5.

En la foto 6 se puede apreciar el resultado del footer tras este cambio.

```
<div class="col-sm-7 footer-copyright wow fadeIn">
  <p>Copyright 2020 Andia - <a href="about.html">About</a> - <a href="contact.html">
  Contact</a> - <a href="RGPD.html">Política de protección de datos</a> - <a href="
  COOKIES.html"> Política de privacidad</a> - <a href="AVISO-LEGAL.html">Aviso legal</a>
  </p>
</div>
```

Foto 5 Añadido pie de página

Copyright 2020 Andia - [About](#) - [Contact](#) - [Política de protección de datos](#) - [Política de privacidad](#) - [Aviso legal](#)

Foto 6 Footer de la página

Lo último que tenemos que hacer es crear el popUp de las cookies. Para ello hemos diseñado un pequeño cuadrado con texto en su interior y un botón de aceptar y una cruz para poder salir.

El resultado del PopUp se muestra a continuación.



Foto 7 Pop-up Cookies

Puesto que solo hay que hacer maquetación del popUp, hemos decidido que tanto si se pulsa el botón aceptar, como si se pulsa la cruz, el popUp se cierre y se muestre la pestaña correspondiente.

Para simplificar la fluidez de la navegación entre pestañas, hemos decidido que este popUp únicamente aparezca en el Home y siempre al recargar la página.

Para realizar esta funcionalidad hemos creado un archivo js llamado popUp\_cookies.js que contiene un script donde a través de la función *addEventListener* vamos dictaminando que deben hacer dichas clases del html al hacer click sobre ellas o al recargarlas.

## Eliminar l'índex antic (netejar repositori). Commitejar al repositori (Developer 2).

Eliminamos el archivo index.html del repositorio local (**git rm index.html**).

IMPORTANTE: Antes de commitear debemos de quitar del archivo .gitignore la carpeta assets, ya que hemos añadido un archivo css y un archivo js a dicha carpeta, correspondientes a la implementación del popUp de las cookies.

Una vez hecho esto, indicamos que archivos queremos que se actualicen en el repositorio remoto (**git add .**) y commiteamos los cambios (**git commit -m "Nou index aplicat a totes les planes, Creat al peu les planes de RGPD (Reglament general de protecció de dades) Política de privacitat / Política de cookies (amb Pop-up de configuració) i Avis legal i eliminat index antic"**).

```
ANGEL@ANGELPC MINGW64 ~/Desktop/p2GDS (ED-B)
$ git commit -m "Nou index aplicat a totes les planes, Creat al peu les planes
de RGPD (Reglament general de protecció de dades) Política de privacitat / Polít
ica de cookies (amb Pop-up de configuració) i Avis legal i eliminat index antic"
[ED-B ff871ca] Nou index aplicat a totes les planes, Creat al peu les planes de
RGPD (Reglament general de protecció de dades) Política de privacitat / Polític
a de cookies (amb Pop-up de configuració) i Avis legal i eliminat index antic
143 files changed, 39325 insertions(+), 87 deletions(-)
```

Foto 8 git commit

## Agregar al (master) i pujar la nova versió al repositori remot (Developer 2)

Para empezar nos movemos a la rama master (**git checkout master**). A continuación hacemos un *merge* en local (**git merge ED-B**) y subimos al repositorio remoto (**git push origin master**).

```
ANGEL@ANGELPC MINGW64 ~/Desktop/p2GDS (master)
$ git merge ED-B
Updating 4e040ef..ff871ca
Fast-forward
 .gitignore | 1 -
 AVISO-LEGAL.html | 169 +
 COOKIES.html | 169 +
 RGPD.html | 169 +
 about.html | 8 +-
 assets/bootstrap/css/bootstrap-theme.css | 476 +
 assets/bootstrap/css/bootstrap-theme.css.map | 1 +
 assets/bootstrap/css/bootstrap-theme.min.css | 5 +
 assets/bootstrap/css/bootstrap.css | 6584 ++++++
```

Foto 9 git Merge ED-B

## Pujar la branca (ED-B) per seguretat (Developer 2)

Para subir la rama ED-B al repositorio remoto, primeramente nos movemos a dicha rama (**git checkout ED-B**) y subimos los cambios al repositorio remoto (**git push origin ED-B**).

**Descarregar modificacions realitzades pel (Developer 2). El client demana una nova modificació, al peu, agregant un camp de subscripció. Crear una nova branca (ED-C) ja que el client no ho té clar... Afegir canvi al repositori remot i ensenyar al client. Es publica al (master) (Developer 1)**

Lo primero que debemos de hacer será descargarnos los cambios realizados por el Developer2, así que nos los descargamos (**git pull origin master**).

A continuación, procedemos a crear la rama ED-C en local (**git branch ED-C**). Acto seguido nos movemos a dicha rama (**git checkout ED-C**), y la subimos al repositorio remoto (**git push origin ED-C**).

Una vez creada la rama, añadimos un botón de suscripción en el footer de la página en todos los archivos html de nuestro proyecto. El resultado visual es el que podemos apreciar en la imagen.

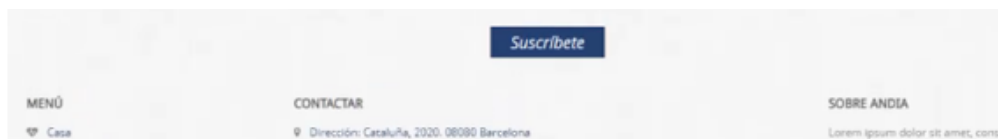


Foto 10 Botón de suscribirse

Para añadir el botón de suscripciones, simplemente añadimos un pequeño código (Foto 11) en todos los archivos.

```
<div class="call-to-action-button">
  <a class="big-link-3" style="margin-top: 40px" href="#">Suscríbete</a>
</div>
```

Foto 11 Código del botón



Una vez realizada esta pequeña modificación en todos los html de nuestro proyecto, procedemos a subir los cambios a la rama master del repositorio remoto.

Una vez hayamos hecho el (`git add .`) para añadir los archivos, estemos en la rama master (`git checkout master`), y hayamos hecho el commit:  
(`git commit -m "camp de subscripcio afegit"`), procedemos a hacer un merge de la rama ED-C a master.

Una cosa que hemos de haber realizado antes de hacer el 'add', es volver a añadir en el archivo `.gitignore` la carpeta `assets/`, puesto que ya no será necesario tenerla en cuenta al no haber realizado ningún cambio significativo con ella.

Como se puede apreciar en la imagen de la derecha, en todos los archivos html se han añadido 3 líneas, las cuales son las correspondientes a las 3 líneas de código ilustradas anteriormente.

```
dc30@LAPTOP-IE9FI5GT MINGW64 ~/works
$ git merge ED-C
Updating ff871ca..4c7342e
Fast-forward
 .gitignore           | 1 +
 AVISO-LEGAL.html     | 3 +++
 COOKIES.html         | 3 +++
 RGPD.html            | 3 +++
 about.html           | 3 +++
 contact.html         | 3 +++
 index-2.html         | 3 +++
 portfolio.html       | 3 +++
 pricing-tables.html  | 3 +++
 services.html        | 3 +++
 10 files changed, 28 insertions(+)
```

Foto 12 git Merge ED-C

**El client acaba dient de treure el "suscríbete" però guardarem la branca (ED-C) amb el canvi per si es repensa... Guardem la branca al repositori remot per seguretat.**

Primero de todo haremos la subida de seguridad de la rama ED-C al repositorio remoto. Para ello, primeramente nos situaremos en la rama ED-C (**`git checkout ED-C`**). Acto seguido subimos los cambios que hay en esta rama al repositorio remoto (**`git push origin ED-C`**).

Aunque hayamos realizado este paso primero, se podría haber realizado posteriormente, tras haber realizado las modificaciones en la rama master.

Para quitar el botón de suscribirse podemos hacerlo de dos maneras: directamente desde la rama master o hacerlo desde la rama ED-C y posteriormente hacer un merge.

Nosotros hemos optado por la primera opción ya que creemos que será más 'limpia' y así no modificamos el código de ED-C.

Para quitar el botón de suscribirse, nos movemos a la rama master (**`git checkout master`**).

Posteriormente borramos de todos los archivos html de nuestro proyecto las 3 líneas donde creamos el botón de suscripción.

Acto seguido, añadimos los cambios al commit (**`git add .`**), commiteamos los cambios (**`git commit -m "boton suscribete quitado"`**), y finalmente subimos los cambios (**`git push origin master`**).

```
dc30@LAPTOP-IE9FI5GT MINGW64 ~/workspace/
$ git commit -m "boton suscribete quitado"
[master ff636a5] boton suscribete quitado
9 files changed, 27 deletions(-)
```

Foto 13 Quitamos el botón en master

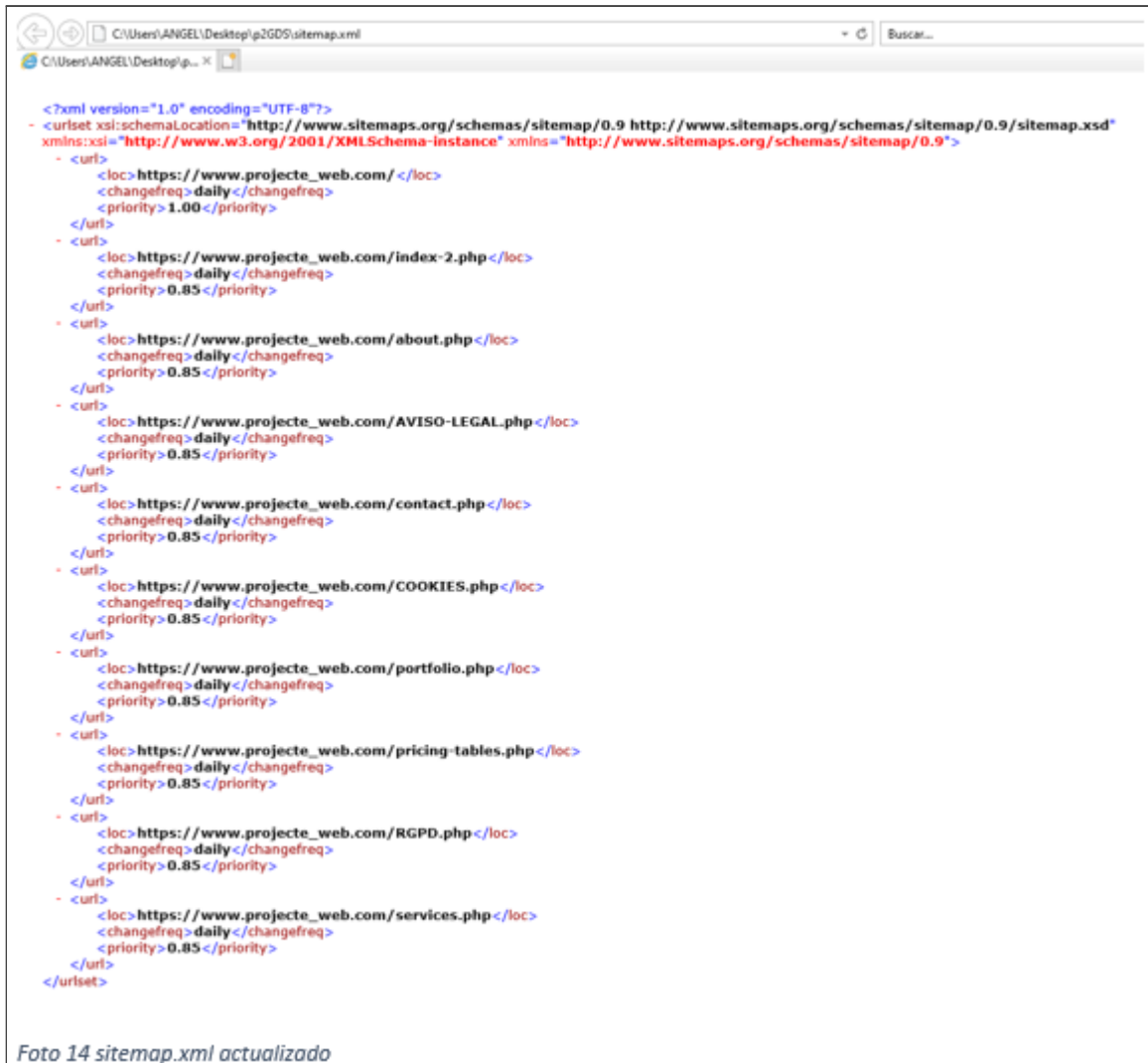
**Descargar i crear Mapa Web, agregar a totes les planes, i el sitemap.xml i publicar al (master) (Developer 2).**

Para este punto primero nos descargamos los cambios (**git pull origin master**).

Además, quitamos el sitemap.xml del .gitignore ya que vamos a modificarlo.

Para realizar esta parte vimos algunos ejemplos de sitemap de páginas reales. Como nuestro trabajo no es una página “real”, decidimos copiar la estructura de esos ejemplos adaptándolo a nuestro caso.

Para finalizar lo añadimos (**git add .**), hacemos un commit (**git commit -m “sitemap.xml actualizado”**) y lo subimos (**git push origin master**).



## Descargar última versió i agregar una pestanya d'ubicació amb l'API de MAPS (Developer 1).

Como siempre, descargamos los cambios realizados por el otro desarrollador (**git pull origin master**).

Primero añadimos a todas las páginas existentes un nuevo panel en el menú de arriba con su enlace a la nueva pestaña de ubicación.

```
<li>
  <a href="ubicacion.html"><i class="fa fa-search"></i><br>Ubicacion</a>
</li>
```

Foto 15 Código del menú superior

Lo siguiente es crear dicho html con el mapa. Simplemente copiamos una de las páginas y la adaptamos a lo necesario.

Utilizamos, siguiendo uno de los pdfs del campus virtual, leafletjs.com.

Con el siguiente código introducimos un mapa dentro de la página, desde el cual se puede navegar por el mundo.

Finalmente lo añadimos al commit (**git add .**), hacemos el commit (**git commit -m "Pestaña ubicación con API MAPS."**) y lo subimos (**git push origin master**).

```
<div class="page-title-container">
  <div class="container">
    <div class="row">
      <div class="col-sm-12 wow fadeIn">
        <i class="fas fa-dot-circle"></i>
        <h1>Ubicacion</h1>
      </div>
    </div>
  </div>
</div>
<!-- Contact Us -->
<div class="contact-us-container">
  <div class="container">
    <div class="row">
      <div id="map">
      </div>
      <script type="text/javascript">
//Cargando nuestro mapa
var mapsipe = L.map('map').
setView([38.6202, -0.5731],
15); // [38.6202, -0.5731] es la latitud y longitud de la zona que queremos mostrar, en nuestro caso
Ibi
L.tilelayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  attribution: 'Map data &copy; <a href="http://openstreetmap.org">OpenStreetMap</a> contributors,
  <a href="http://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, Imagery © <a
  href="http://cloudmade.com">CloudMade</a>',
  maxZoom: 18
}).addTo(mapsipe);
</script>
```

Foto 16 Código del mapa

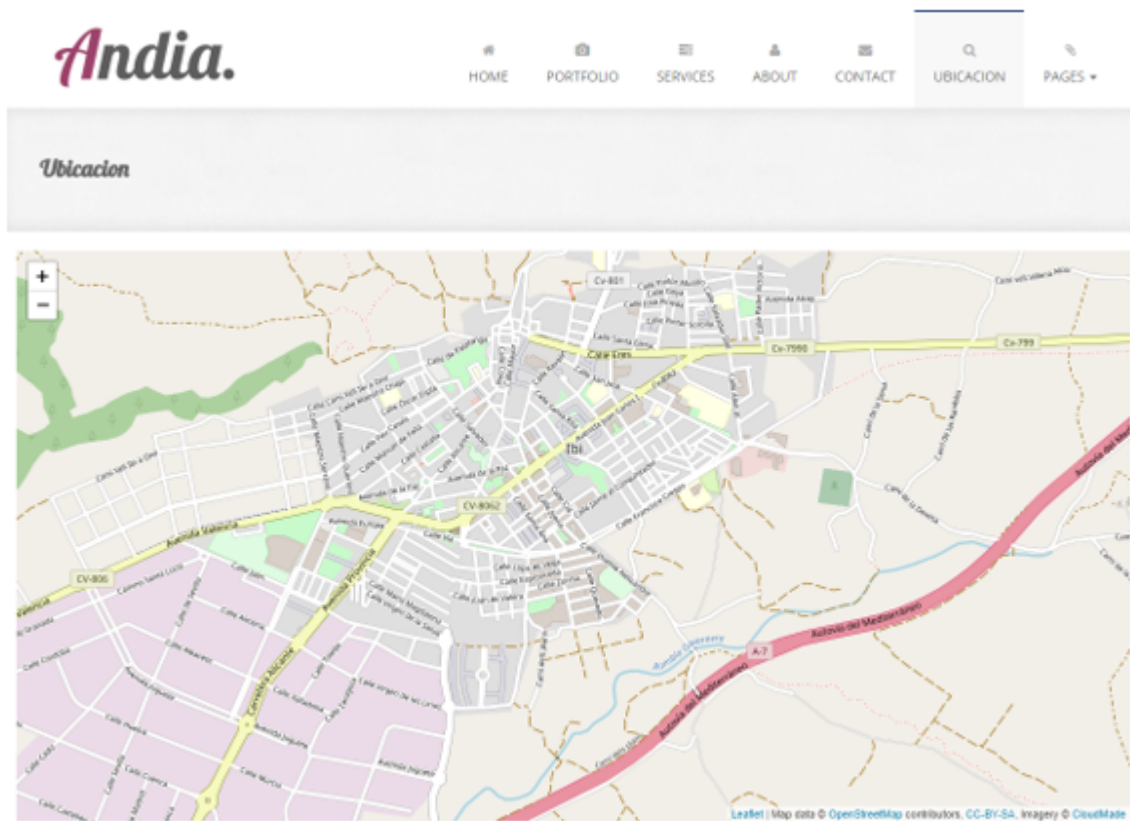


Foto 17 Página de ubicación

## Publicar i donar accés al professor a Bitbucket [ >> ].

Como última tarea, damos acceso al profesor desde el mismo BitBucket, mandando una invitación con el permiso de lectura.



Foto 18 Dar acceso al profesor

# FINALIZACIÓN

Desde la pestaña de commits del repositorio podemos observar todos los commits realizados a lo largo de este trabajo. Y si hacemos click en uno de ellos se podrán ver los cambios realizados en ese paso del trabajo.

Daniel Calvo Ramos / PRACTICA2\_GDS / ProjecteWeb

## Commits








Search commits		All branches		
Author	Commit	Message	Date	
 Daniel Calvo Ramos	<a href="#">7336454</a>	Pestaña ubicacion con API MAPS.	1 minute ago	
 Angel Garcia Calleja	<a href="#">d115be6</a>	sitemap.xml actualizado	28 minutes ago	
 Daniel Calvo Ramos	<a href="#">ff636a5</a>	boton suscribete quitado	48 minutes ago	
 Daniel Calvo Ramos	<a href="#">4c7342e</a>	camp de subscripcio afegit	1 hour ago	
 Angel Garcia Calleja	<a href="#">ff871ca</a>	Nou index aplicat a totes les planes, ...	1 hour ago	
 Daniel Calvo Ramos	<a href="#">4e040ef</a>	Base ProjecteWeb	2 hours ago	
 Daniel Calvo Ramos	<a href="#">6887557</a>	.gitignore creado	2 hours ago	

Foto 19 Commits del repositorio

Como conclusión podemos decir que esta práctica nos ha ayudado a aprender a manejar el uso de repositorios y git, herramientas que tendremos que utilizar seguro en un futuro. Hemos observado el uso de ramas para no “pisarnos” entre desarrolladores.