

**ACTIVITAT AVALUABLE AC8****Mòdul:** MP08- Desplegament d'aplicacions web**UF:** UF1 – Servidors web i de transferència de fitxers**Professor:** Albert Guardiola**Data límit d'entrega:** 24/11/2024**Mètode d'entrega:** Per mitjà del Clickedu de l'assignatura. Les activitats entregades més enllà de la data límit només podran obtenir una nota de 5.**Instruccions:** S'ha d'entregar un únic document amb el nom:***MP08-UF1-AC8-Nom_Alumne.doc (o pdf)***

Es valorarà la presentació.

Resultats de l'aprenentatge:

RA1. Implanta arquitectures web analitzant i aplicant criteris de funcionalitat.

RA2. Gestiona servidors web avaluant i aplicant criteris de configuració per a l'accés segur als serveis.

Tasques a realitzar:

Creació dels repositoris necessaris per a la pràcticaTasca 1 a) Crea en el teu compte personal de *GitHub* dos repositoris per albergar, respectivament, els projectes *m8ac7* i *m8ac7_python*. Penja-hi el codi adjunt.b) **Assegura't que els dos projectes estan configurats com privats.**

Des d'agost de 2021, l'accés remot (mitjançant HTTPS) als repositoris de *github.com* ja no s'acredita amb nom i contrassenya, sinó que és necessari un *token* al qual s'associen uns privilegis determinats des de l'administració del repositori:

<https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls>

Donat que hem configurat els repositoris com a privats, **haurem de generar un *token* personal per a poder fer-los servir:**

<https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>



c) Crea el teu *token* personal seguint les instruccions de l'enllaç anterior. Assegura't que el configures per a tenir accés només als dos repositoris que acabes de crear, amb permisos de "Contents". El resultat hauria de ser quelcom així:

test

No description

Created yesterday.

Expires on Sun, Dec 11 2022.

Access on albertetpx

Repository access

- albertetpx/m8ac7
- albertetpx/m8ac7_python

User permissions

This token does not have any user permissions.

Repository permissions

- ✓ Read access to metadata
- ✓ Read and Write access to code

Un cop l'assistent et mostri el teu *token*, és necessari que el copiis en un lloc segur. Github no te'l tornarà a mostrar, per raons de seguretat. Si et cal veure'l un altre cop, demana que te'l regeneri.

Instal·lació de control de versions al servidor

És imprescindible tenir un control de la versió de codi que es desplega en un servidor de producció (en el nostre cas, el que tenim instal·lat a la màquina virtual). L'eina GIT ens permet aconseguir-ho de manera senzilla.

Tasca 2. a) Instal·la GIT en el servidor de la màquina virtual:

```
sudo apt-get update
```

```
sudo apt install git
```



c) Al directori `/var/www/html`, clona el teu projecte remot `m8ac7`:

```
sudo git clone https://github.com/<usuari-github>/m8ac7.git
```

(posa com a nom el teu usuari de *GitHub* i com a contrassenya el *token* generat a la tasca anterior. Si tens problemes per introduir el token quan el CLI et demani la contrassenya, pots afegir-la a la comanda clone:

```
sudo git clone
```

```
https://<usuari-github>:<token>@github.com/<usuari-github>/m8ac7.git
```

d) Demana el nou recurs del servidor a <http://<ip-màquina-virtual>/m8ac7>.

Instal·lació de PHP al servidor

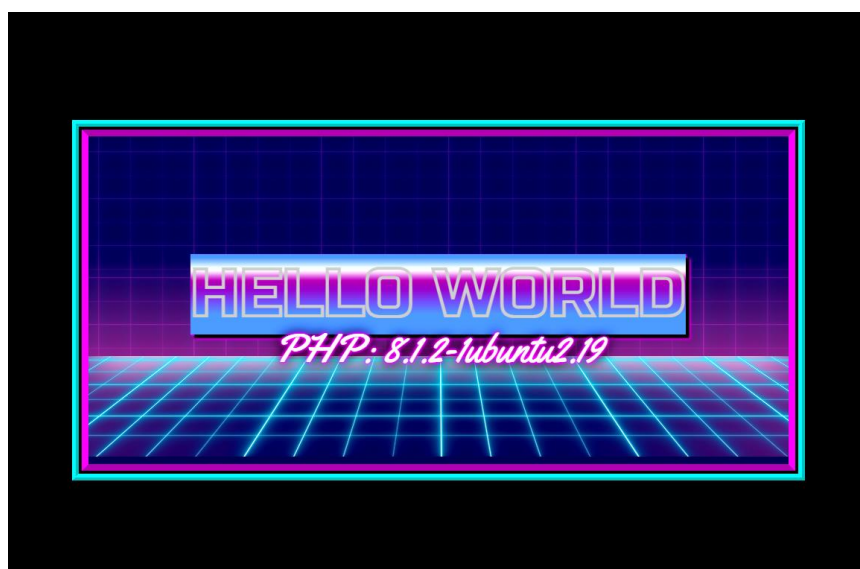
A la web anterior hauria d'aparèixer la versió instal·lada de PHP. Previsiblement no serà així: haurem d'instal·lar prèviament PHP al servidor, així com el mòdul que el connecta a Apache.

Tasca 3. a) Instal·la els mòduls necessaris:

```
sudo apt install php php-mysql libapache2-mod-php
```

b) Assegura't que, com a part del procés d'instal·lació, s'ha habilitat el mòdul necessari per a que Apache pugui servir pàgines PHP. Observa els nous mòduls *php-xxx* presents al directori *mods-available*.

c) Després de reiniciar el servidor, demana el nou recurs del servidor a <http://<ip-màquina-virtual>/m8ac7> i comprova que el codi PHP s'ha pogut executar amb èxit: **s'hauria de mostrar la versió de PHP instal·lada:**





Tasca 4. **Ara farem servir la nostra màquina amfitriona com a màquina de desenvolupament.**

a) Desde VSCode, clona el projecte <https://github.com/<usuari-github>/m8ac7.git>: assegura't que el projecte es clona a dins de la carpeta *htdocs* de XAMPP, ja que necessita un motor de PHP per executar-se. b) Demana la web al servidor local desde el navegador.

c) Fes algun canvi al codi, comprova'ls des del navegador local, i puja els canvis al repositori remot (*commit + sync*).

d) Actualitza't el projecte al servidor de producció (màquina virtual: *sudo git pull*) i observa com s'actualitza la web servida a la IP del servidor: <http://<ip-màquina-virtual>/m8ac7>.

Tasca 5. **Ja tenim la màquina de producció i la màquina de desenvolupament enllaçades en un mateix workflow, gràcies al repositori remot. Ens queda un element important per tenir l'entorn de desenvolupament complet: un depurador (*debugger*).**

Farem servir el depurador *Xdebug* que ve per defecte instal·lat amb XAMPP.

a) Per habilitar-lo, és necessari seguir amb cura les instruccions que dona la pàgina www.xdebug.org.

Xdebug ha desenvolupat un assistent per facilitar el procés d'instal·lació:

<https://xdebug.org/wizard>

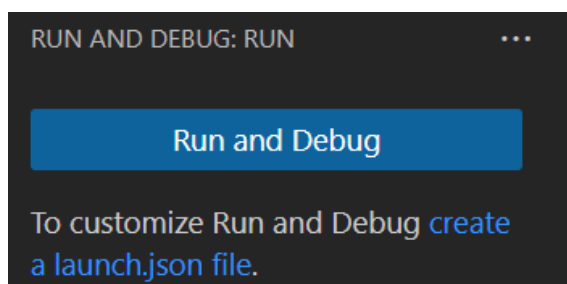
Tant si tens l'extensió XDEBUG instal·lada a XAMPP (*xampp/php/ext*) com si no, copia en l'assistent la informació de sortida del PHPInfo. La trobaràs al menú PHPInfo del *dashboard* de XAMPP:

<http://localhost/dashboard/phpinfo.php>

L'assistent us indicarà quina versió de la llibreria (.dll) de xdebug correspon a la vostra versió de PHP i quines directives heu d'afegir al fitxer *php.ini* (*xampp/php/php.ini*) per a activar-la.

b) A VSCode, instal·leu l'extensió *PHPDebug* (de *Xdebug*).

A continuació, amb el projecte *m8ac7* obert, des de la pestanya lateral de depuració, seleccioneu l'opció *create launch.json file*: se us obrirà el fitxer de configuracions de depuració. Observeu que hi ha tres possibles configuracions de depuració: *Listen for xdebug*, *Launch currently open script* i *Launch built-in web-server*.

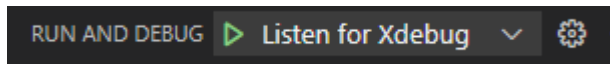


En aquesta pràctica, **només farem servir la configuració *Listen for Xdebug*: l'execució del codi s'iniciarà remotament (quan es demani una web al servidor XAMPP), però el nostre podrà**



“**escoltar**” pel port de xdebug, de manera que tindrà un control d’aquesta execució. Això permetrà posar *breakpoints*, prendre el control de l’execució, inspeccionar el valor de les variables, i alliberar de nou l’execució del codi a mans del servidor.

c) Poseu un breakpoint en alguna línia de PHP, i inicieu l’escolta del port de Xdebug, a la pestanya de depuració del projecte.



d) Demaneu des del navegador el recurs <http://localhost/m8ac7...> S’hauria de parar l’execució del codi PHP un cop es detecti que es passa pel *breakpoint*. Comproveu que VSCode pren el control de l’execució... i que la podeu reanudar a voluntat.