



UNIVERSIDAD DE PANAMÁ
CENTRO REGIONAL UNIVERSITARIO DE VERAGUAS
FACULTAD DE INFORMÁTICA, ELECTRÓNICA Y
COMUNICACIÓN

JORNADA DE ACTUALIZACIÓN ACADÉMICA CRUV-FIEC-
2021

Expositores

- Ing. Emeldo Quirós: ***Ciclo de vida del software: SCRUM y DevOps***
- Lic. Moisés Castillo: ***Configuración de un entorno de desarrollo.***

Recopilado por
Carlos J. Concepción P.

Jornada realizada desde
El 8 al 12 de febrero de 2021

- ***Extracto analítico y reflexivo sobre el Ciclo de Vida: SCRUM y DevOps.***

En el ámbito de desarrollo de software, existen metodologías que se implementan para llevar a cabo una buena gestión de proyecto, con la finalidad de presentar un producto con un valor mayor, y de excelente calidad.

Las metodologías ágiles son unas de las mentoras utilizadas en muchas empresas, para el desarrollo de sistemas de las mismas, ya que fomentan el desarrollo ágil del software en cuestión, enfocándose en requisitos y soluciones evolutivas dependiendo de las necesidades que se vayan presentando en el proyecto.

La metodología Scrum, parte desde la idea de desarrollar proyectos implementando las mejores prácticas posibles, con el fin de enfocarse en la necesidad de valor que desea el usuario o cliente final. Los valores de Scrum van ligados a la idea de que las personas involucradas en el desarrollo, tengan la convicción de que se puede llegar a obtener un buen resultado, y así poder entregar un producto de calidad. Además, crea el entorno de tener compromisos con el proceso de entrega del trabajo, estimulando la idea de que se puedan equivocar, pero que en el marco del respeto se puedan ayudar entre sí.

Scrum establece reuniones o equipos en los cuales se manejan los requisitos que se deben tomar en cuenta en el desarrollo. Lo primero que se establece es el inception o comienzo, en donde se listan los alcances de que se quieren hacer, se ve la disponibilidad de los recursos, se generan los documentos de inicio, entre otros. Lo segundo es el sprint o punto de ejecución, que es donde se establecen las revisiones y definen lo que se hará en las siguientes, para evitar omitir tomar en cuenta algo que puede ser importante más adelante.

En conclusión, en cuanto al Scrum, siempre se deben definir requisitos o normas que hagan que el proyecto en desarrollo de la empresa sea efectivo y muy amigable. En las reuniones que aquí se contemplan, se habla del inicio o introducción a como se pueden o se van a realizar las tareas. Se analizan todos los recursos disponibles para luego establecer la reunión (sprint) en donde se detallan como van



Carlos Concepción

evolucionando las actividades definidas previas al desarrollo del proyecto. Lo importante es tener un avance al final de cada sprint; para tratar de tener siempre un producto terminado.

Por otra parte, DevOps, busca entre otras cosas crear una conjugación directa entre todas las personas que tienen que ver con el desarrollo del software. Esto se hace con el fin de que las personas que realizan las operaciones y los desarrolladores puedan interactuar de manera directa y colaborar a lo largo del proceso, tratando de omitir la creación de hilos de trabajos individuales diferentes; es decir, evitar que los desarrolladores de software trabajen de manera individual a las personas que ejecutan las operaciones.

La forma en que se da todo este sistema de interacción, provoca que el software sea entregado a nivel de infraestructura; es decir, lo que busca DevOps es que cuando el software está listo, la infraestructura esté disponible para un producto desde el día de inicio del desarrollo, para poder publicarlo y así evitar que cuando se termine el desarrollo, estar ocupado buscando donde se puede publicar.

El enfoque generado por DevOps viene concurrido por la idea de unificar a todo el equipo, el de seguridad, infraestructura y demás, acogidos todos en un solo proceso en el desarrollo del software. Por eso se dice que DevOps es una cultura en la que unidos todos como uno, se da disponibilidad a todos los recursos necesarios para el producto.

El tema del Scrum queda dentro del contexto DevOps, pero sigue siendo el mismo proceso. Se dan las revisiones dentro del proceso DevOps, eliminando los hilos y aumentando a un nivel mayor la comunicación. Con esta colaboración Scrum-DevOps, las personas se dan cuenta más rápido de las fallas que van surgiendo a lo largo del desarrollo y así aprenden de las cosas que se hacen y como se hacen.

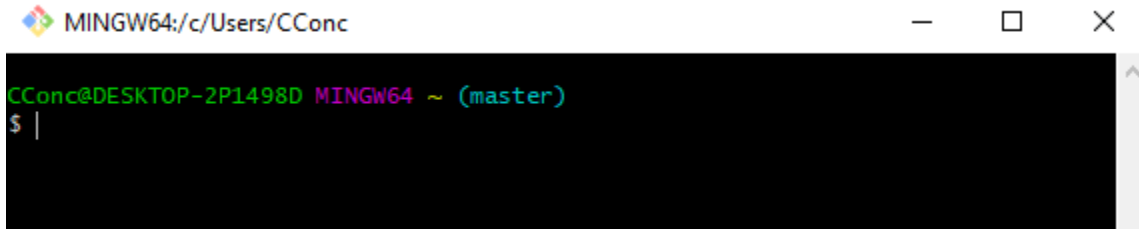


Carlos Concepción

- **Evidencia de la realización de las prácticas acerca de Entornos de Desarrollo.**

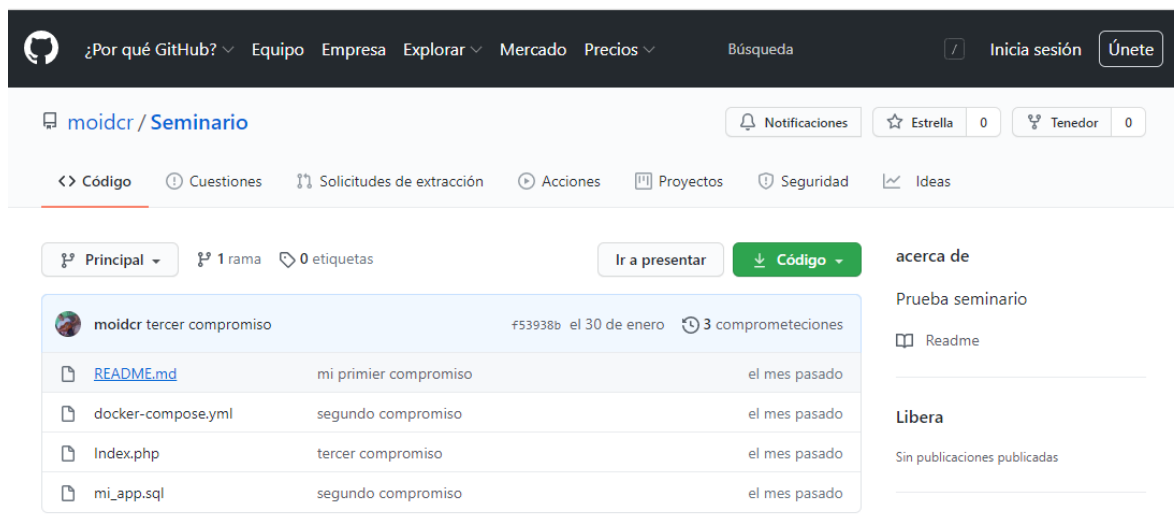
CLONANDO REPOSITORIOS CON GIT

Antes de empezar a clonar repositorios, debemos descargar e instalar Git. Luego ejecutamos el Git Bash:



Seguidamente, para clonar un repositorio o archivos con git necesitamos el url o ruta donde se encuentra alojado, en este caso utilizaremos Github, la ruta la obtendremos en el repositorio deseado.

La ruta en este caso sería: <https://github.com/moidcr/seminario.git>



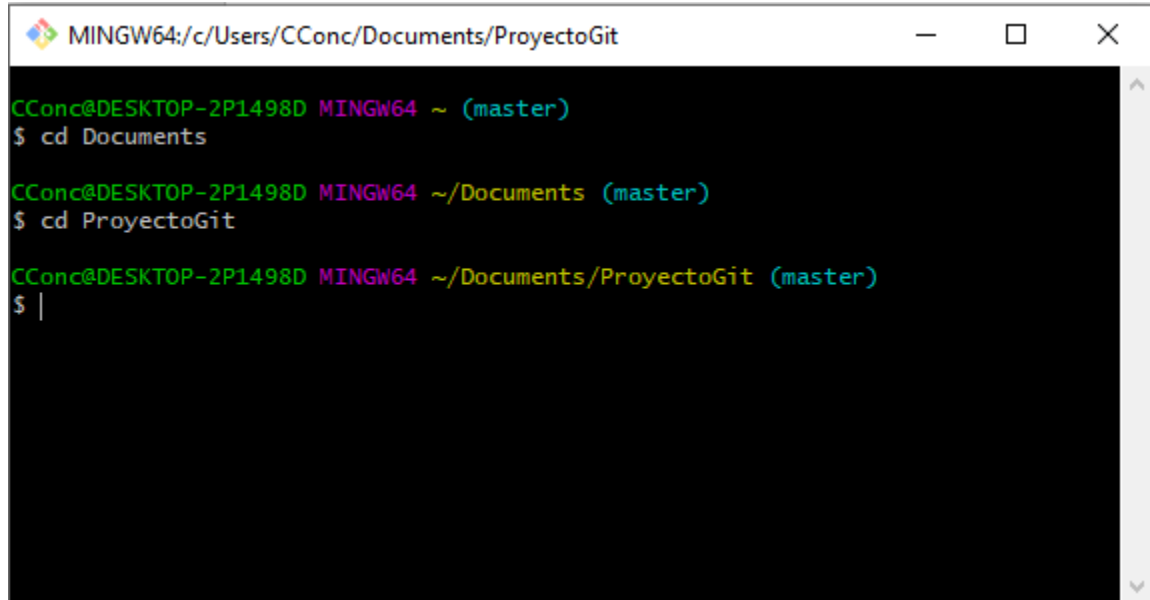
Ahora utilizando el Git Bash, nos dirigiremos a una ubicación que nos resulte sencilla para clonar los archivos. En mi caso, lo haré en la siguiente ubicación:

C:\Users\CConc\Documents\ProyectoGit



Carlos Concepción

Lo cual en el Git Bash, sería así:



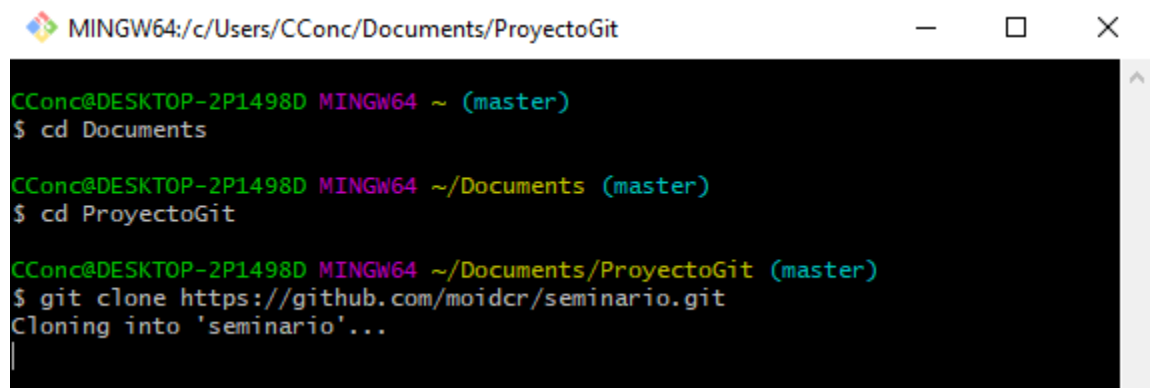
```
MINGW64:/c/Users/CConc/Documents/ProyectoGit
CConc@DESKTOP-2P1498D MINGW64 ~ (master)
$ cd Documents

CConc@DESKTOP-2P1498D MINGW64 ~/Documents (master)
$ cd ProyectoGit

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit (master)
$ |
```

Estando ubicado en el archivo donde deseamos clonar el archivo de GitHub, ejecutamos en siguiente comando:

git clone <https://github.com/moidcr/seminario.git>



```
MINGW64:/c/Users/CConc/Documents/ProyectoGit
CConc@DESKTOP-2P1498D MINGW64 ~ (master)
$ cd Documents

CConc@DESKTOP-2P1498D MINGW64 ~/Documents (master)
$ cd ProyectoGit

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit (master)
$ git clone https://github.com/moidcr/seminario.git
Cloning into 'seminario'...
```

Esperamos hasta que culmine el clonado de los archivos.



Carlos Concepción

Al culminar de clonar los archivos, se debe reflejar la siguiente pantalla.

```

MINGW64; c:/Users/CConc/Documents/ProyectoGit
CConc@DESKTOP-2P1498D MINGW64 ~ (master)
$ cd Documents

CConc@DESKTOP-2P1498D MINGW64 ~/Documents (master)
$ cd ProyectoGit





CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit (master)
$ git clone https://github.com/moidcr/seminario.git
Cloning into 'seminario'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 11 (delta 2), reused 11 (delta 2), pack-reused 0
Unpacking objects: 100% (11/11), 2.57 KiB | 5.00 KiB/s, done.

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit (master)
$

```

Y así, nos quedaría la carpeta **C:\Users\CConc\Documents\ProyectoGit\seminario**, luego de clonar los archivos de la dirección **git clone <https://github.com/moidcr/seminario.git>**.

Este equipo > Disco local (C:) > Usuarios > CConc > Documents > ProyectoGit > seminario

<input type="checkbox"/> Nombre	Fecha de modificación	Tipo	Tamaño
 docker-compose.yml	03/10/2021 7:07 p. m.	Archivo YML	2 KB
 index	03/10/2021 7:07 p. m.	Archivo PHP	3 KB
 mi_app	03/10/2021 7:07 p. m.	SQL Text File	1 KB
 README.md	03/10/2021 7:07 p. m.	Archivo MD	1 KB

Y mostrando desde Git Bash, nos quedaría:

```

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit (master)
$ cd seminario

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ ls
README.md  docker-compose.yml  index.php  mi_app.sql

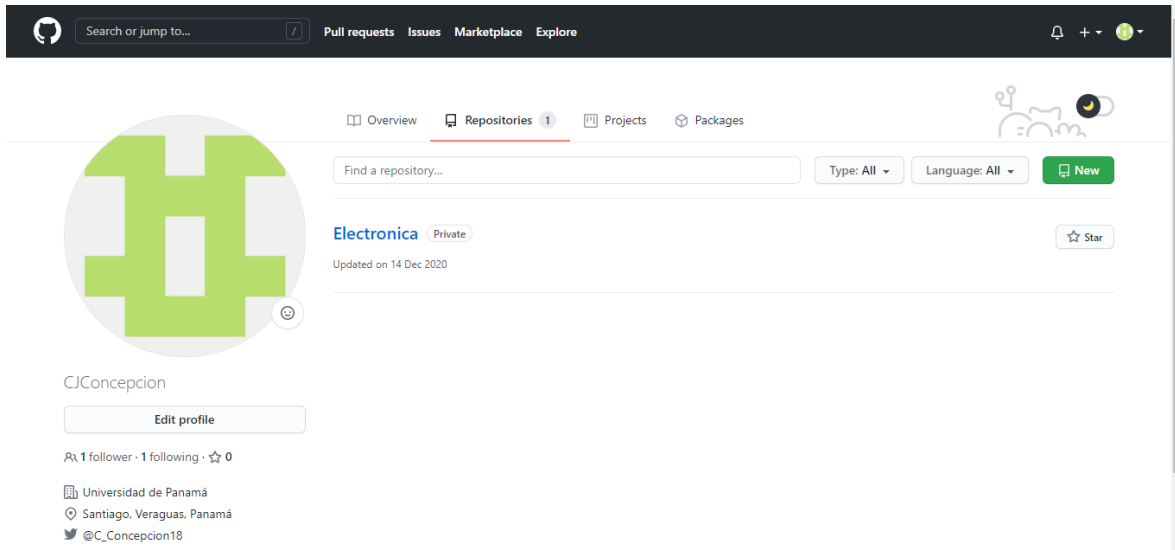
CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$

```



Carlos Concepción

Ahora, para subir archivos a un repositorio en GitHub tenemos que registrar una cuenta de GitHub. En mi caso, ya tenía una cuenta personal registrada, por lo que sólo tuve que subir los archivos.



Ahora utilizando el Git Bash nos ubicamos dentro del directorio seminario, y ejecutamos los siguientes comandos para subir nuestro repositorio directorio a GitHub:

```
MINGW64:/c:/Users/CConc/Documents/ProyectoGit/seminario

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit (master)
$ cd seminario

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ |
```

Luego de ubicarnos dentro del directorio seminario, inicializamos Git en nuestro directorio:



```
MINGW64:/c/Users/CConc/Documents/ProyectoGit/seminario
CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit (master)
$ cd seminario

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git init
Reinitialized existing Git repository in C:/Users/CConc/Documents/ProyectoGit/seminario/.git/

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ |
```

Seguidamente agreguemos con un punto (.) todos los archivos al **staging** área:

```
CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git add .

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ |
```

Colocamos nuestro proyecto en el repositorio local con nuestro primer **commit**:

```
CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git commit -m "Mi primer Commit"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ |
```

Ahora, definimos la rama de trabajo:

```
CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git branch -M main

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ |
```

Indiquemos la ruta a nuestro repositorio Git, si obtenemos un error se trata de que sigue apuntando al repositorio de procedencia, cambiémoslo con el **set-url**:





```
CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git remote add origin https://github.com/CJConcepcion/seminario1.git
fatal: remote origin already exists.

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git remote set-url origi https://github.com/CJConcepcion/seminario1.git
fatal: No such remote 'origi'


CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git remote set-url origin https://github.com/CJConcepcion/seminario1.git


CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ |
```

Y por último realizamos el **push** al repositorio indicado, indicando nuestras credenciales de GitHub, obteniendo el siguiente resultado:

 GitHub Login ×

GitHub Login

 Login

 Cancel

Don't have an account? [Sign up](#)

[Forgot your password?](#)



```
CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git remote add origin https://github.com/CJConcepcion/seminario1.git
error: remote origin already exists.

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git remote set-url origin https://github.com/CJConcepcion/seminario1.git

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git push -u origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (11/11), 2.58 KiB | 43.00 KiB/s, done.
Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/CJConcepcion/seminario1.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ |
```

Quedando nuestro repositorio en GitHub de la siguiente manera:

The screenshot shows the GitHub interface for the repository 'CJConcepcion/seminario1'. At the top, there are navigation tabs: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below these, there's a section for the repository's main branch, 'main', showing 1 branch and 0 tags. A table lists the repository's files and their commit history:

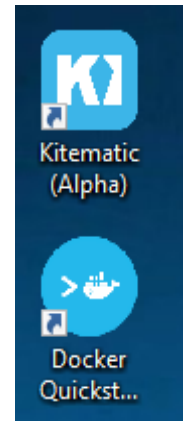
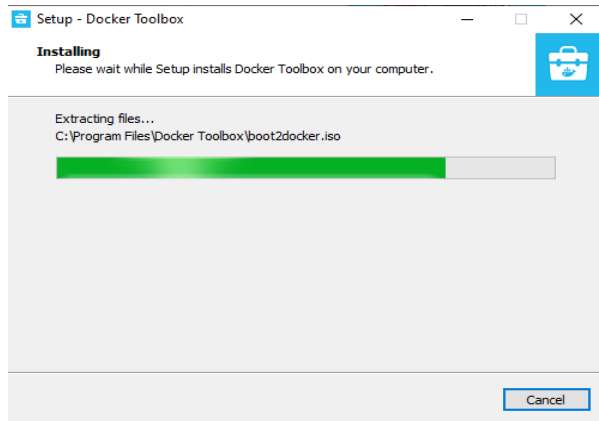
File	Commit	Time
README.md	mi primer commit	last month
docker-compose.yml	second commit	last month
index.php	third commit	last month
mi_app.sql	second commit	last month

Below the table, there's a preview of the README.md file, which contains the word 'seminario'. On the right side, there are sections for 'About', 'Releases', and 'Packages', each with a brief description and a link to create a new release or package.

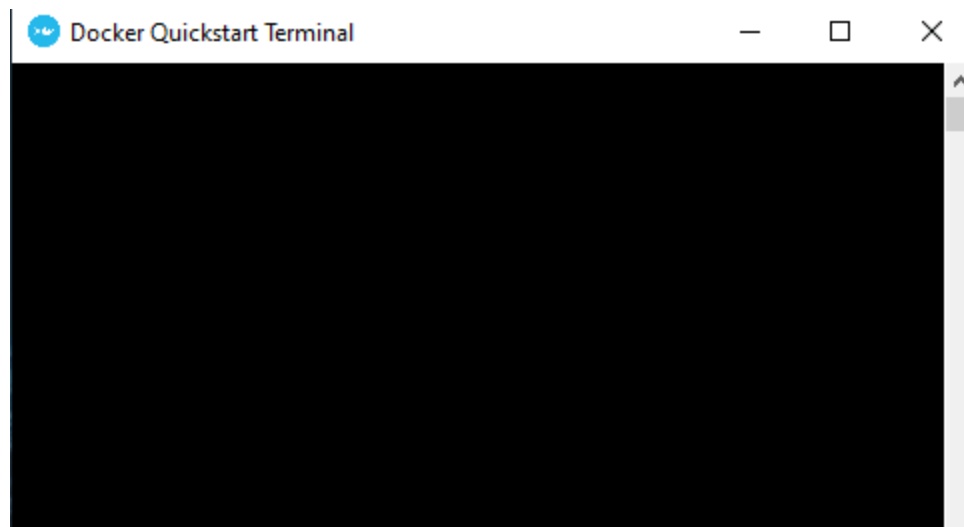


ENTORNO PHP CON DOCKER

Para empezar con Docker en entornos php, debemos hacer la instalación de Docker Toolbox:



Seguidamente, ejecutamos el terminal Docker QuickStar en modo administrador para que no haya problemas de compatibilidad y acceso a los archivos.



Luego de que termina la ejecución, nos aparece Moby (la ballena de Docker), lo que significa que su instalación fue exitosa.



```
MINGW64:/c/Users/CConc
```

docker is configured to use the default machine with IP 192.168.99.100
For help getting started, check out the docs at <https://docs.docker.com>

Start interactive shell

```
CConc@DESKTOP-2P1498D MINGW64 /c/Program Files/Docker Toolbox  
$ cd
```

Necesitamos posicionarnos en la carpeta donde tenemos nuestro archivo ***docker-compose.yml***. En mi caso lo tengo en la dirección ***C:\Users\CConc\Documents\ProyectoGit\seminario***.

```
MINGW64:/c/Users/CConc/Documents/ProyectoGit/seminario

CConc@DESKTOP-2P1498D MINGW64 ~ (master)
$ cd Documents

CConc@DESKTOP-2P1498D MINGW64 ~/Documents (master)
$ cd ProyectoGit

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit (master)
$ cd seminario

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$
```

Estando ubicados en el directorio seminario procederemos a iniciar la instancia de nuestro sistema, para ello ejecutemos en el terminal el comando necesario: ***docker-compose up -d***

```
MINGW64:/c:/Users/CConc/Documents/ProyectoGit/seminario
CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit (master)
$ cd seminario
CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ docker-compose up -d
Creating network "seminario_default" with the default driver
Pulling web (alterway/php:5.4-apache)...
```



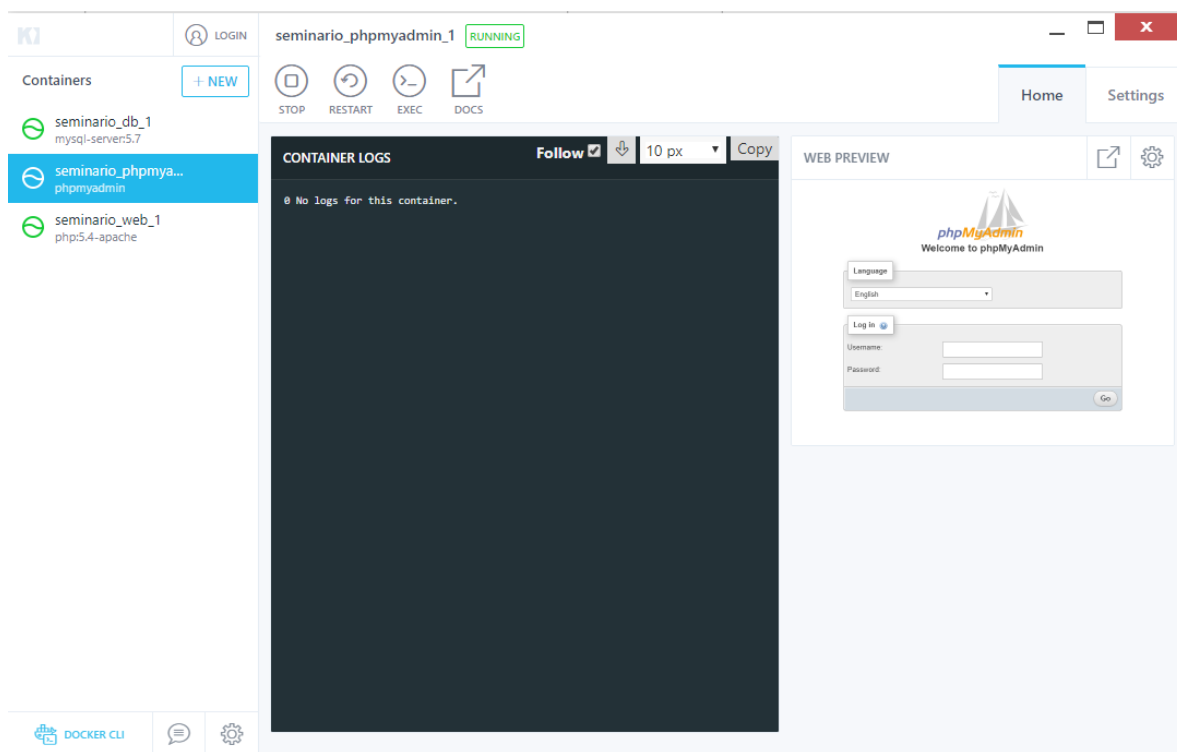
Carlos Concepción

Al finalizar, se nos despliega los contenedores con su respectivos **done** indicando que todo está listo y corriendo sus funcionalidades.

```
Status: Downloaded newer image for phpmyadmin/phpmyadmin:latest
Creating seminario_web_1 ... done
Creating seminario_db_1 ... done
Creating seminario_phpmyadmin_1 ... done

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$
```

Accedemos a **Kitematic**, utilizando los servicios de phpmyadmin y en donde veremos nuestros contenedores.



Utilizando cualquier editor de texto (Sublime Text en mi caso), accedemos al archivo index.php para hacer unos cambios en el código del archivo:



Hola! PHP 5 te saluda desde Docker

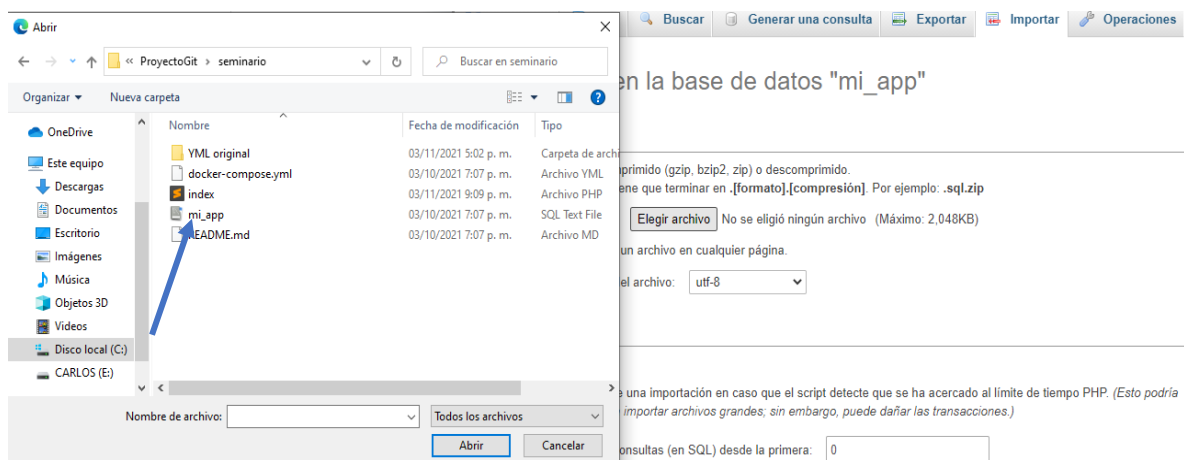
Programa

Utilidad

id	Título	Utilidad
----	--------	----------

(!) Fatal error: Call to a member function fetch_array() on a non-object in /var/www/html/index.php on line 58				
Call Stack				
#	Time	Memory	Function	Location
1	0.0030		235856{main}()	.../index.php:0

Ahora estando en phpmyadmin, importamos el archivo mi_app.sql para inicializar la base de datos:



Luego de importar el archivo, nos arroja que la importación fue exitosa:



192.168.99.100:8081/index.php?route=/import

phpMyAdmin

Base de datos: mi_app

Importación ejecutada exitosamente, 5 consultas ejecutadas. (mi_app.sql)

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0002 segundos.)

```
/* Navicat Premium Data Transfer Source Server : Local Source Server Type : MySQL Source Server Version : 50650 Source Host : localhost:3306 Source Schema : myapp Target Server Type : MySQL Target Server Version : 50650 File Encoding : 65001 Date: 04/01/2021 15:47:04 */ SET NAMES utf8mb4
```

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0004 segundos.)

```
SET FOREIGN_KEY_CHECKS = 0
```

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0005 segundos.)

```
----- Table structure for programas ----- DROP TABLE IF EXISTS `programas`
```

Note: #1051 Tabla 'mi_app.programas' desconocida

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0202 segundos.)

```
CREATE TABLE `programas` ( `id` int(22) NOT NULL AUTO_INCREMENT, `Titulo` varchar(50) NOT NULL, `Utilidad` text NOT NULL, PRIMARY KEY (`id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
```

Esta sería nuestra tabla en phpmyadmin importada:

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0004 segundos.)

```
SELECT * FROM `programas`
```

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

id	Título	Utilidad
----	--------	----------

Operaciones sobre los resultados de la consulta

Crear vista

Y nuestro contenedor web nos quedaría así:

Guardar

id	Título	Utilidad
----	--------	----------

Le hacemos unos pequeños cambios de información a nuestro contenedor web:



Hola! Haciendo mi primer Docker en entornos PHP

Software

Funcionalidad

Salvar

Identificador	Nombre	Funcionalidad
---------------	--------	---------------

Ahora hacemos un commit para subir nuestro repositorio a GitHub:

```
CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git commit -m "Archivos de Docker en entornos PHP"
[main 1b69e51] Archivos de Docker en entornos PHP
2 files changed, 33 insertions(+), 7 deletions(-)
create mode 100644 YML original/docker-compose.yml

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git branch -M main

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git remote add origin https://github.com/CJConcepcion/Docker_en_entornos_php.git
error: remote origin already exists.

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git remote set-url origin https://github.com/CJConcepcion/Docker_en_entornos_php.git

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git push -u origin main
remote: Repository not found.
fatal: repository 'https://github.com/CJConcepcion/Docker_en_entornos_php.git/' not found

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ git push -u origin main
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 4 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (16/16), 3.31 KiB | 91.00 KiB/s, done.
Total 16 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/CJConcepcion/Docker_en_entornos_php.git
* [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

CConc@DESKTOP-2P1498D MINGW64 ~/Documents/ProyectoGit/seminario (main)
$ |
```

Seguidamente, de esta manera visualizamos nuestros archivos Docker en GitHub:



CJConcepcion / Docker_en_entornos_php

Unwatch1Star0Fork0

<> CodeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

main1 branch0 tags

Go to fileAdd fileCode

CJConcepcion

Archivos de Docker en entornos PHP

1b69e51 · 9 minutes ago · 4 commits

YML original

Archivos de Docker en entornos PHP

9 minutes ago

README.md

mi primer commit

last month

docker-compose.yml

second commit

last month

index.php

Archivos de Docker en entornos PHP

9 minutes ago

mi_app.sql

second commit

last month

About

No description, website, or topics provided.

Readme

Releases

No releases published

Create a new release