

## Manual de Instalación

### Planificación Semestral de Asignaturas UTEM

---

| Integrantes              |
|--------------------------|
| Carlos Guerrero Urbina   |
| Daniel Gutierrez Pizarro |
| Ricardo López Núñez      |

|                   |                          |
|-------------------|--------------------------|
| <b>Asignatura</b> | Ingeniería de Software   |
| <b>Profesor</b>   | Sebastián Salazar Molina |
| <b>Fecha</b>      | Lunes 30 de Junio,2014   |

## **Tabla de Contenidos**

|                                     |    |
|-------------------------------------|----|
| Tabla de Contenidos                 | 2  |
| Introducción                        | 3  |
| 1.1 Alcances                        | 4  |
| 1.2 Limitaciones                    | 4  |
| 1.3 Requisitos                      | 5  |
| 1.3.1 Requisitos de Hardware        | 5  |
| 1.3.2 Requisitos de Software        | 5  |
| 1.4 Instalación                     | 6  |
| 1.4.1 Descarga del Programa         | 6  |
| 1.4.2 Instalación del Programa      | 7  |
| 1.4.3 Puesta en Marcha del Proyecto | 8  |
| 1.5 Preguntas Frecuentes            | 10 |
| 1.6 Errores Comunes                 | 11 |

## INTRODUCCIÓN

En una organización de tipo educacional uno de los pilares fundamentales es la docencia y un punto importante a considerar es cómo impartirla, es decir como ordenar y planificar los contenidos de cada clase.

En nuestra Institución se aplican procesos en los cuales los profesores dan a entender cómo administraran el contenido para las clases que ellos dictaran. La Universidad hoy en día hace uso de un formulario el cual ingresa sólo una vez cada vez que se inicia un nuevo semestre una de las desventajas es que no pueden modificarlo durante éste mismo.

Para ello se realizó una aplicación la cual ayuda a resolver este tipo de situación, como también proporcionar un medio eficaz de comunicación con la comunidad universitaria.

En el presente manual se enseñará forma correcta de instalar dicho sistema cumpliendo los requisitos explicados en el manual, además de proporcionar una ayuda a los problemas más frecuentes y cómo solucionarlos.

## 1.1 ALCANCES

Este proyecto abarca una planificación de clases, mediante la cual se puede programar y llevar un registro de las pautas que se impartirán clase a clase .

Para poder iniciar el proyecto fue necesario integrar al cliente con el objetivo de conocer que es lo que necesita que contenga el proyecto .

Dentro de los requerimientos del cliente se desprenden ciertas características que la página debe tener:

- Una interfaz llamativa
- Estar dentro de los presupuestos ( costo )
- Los desarrolladores del proyecto sean eficientes
- Se entregue dentro del periodo estimado
- Fácil de usar y de fácil acceso , es decir, que sea simple navegar dentro de la página

Con respecto al trabajo en equipo , cada integrante realiza sus aportes en conjunto , se realiza un análisis de los mismos y el resultado es llevado a la práctica y subido a github. **GitHub** es una forma para alojar proyectos utilizando el sistema de control de versiones Git. De esta manera se puede llevar un orden en la planificación de la información del proyecto.

## 1.2 LIMITACIONES

Para llevar a cabo la implementación de este proyecto se ocupo Linux como sistema operativo y PHP , que es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

- PostgreSQL es una herramienta que se encarga de crear y guardar información de forma estructurada y que permite el rápido acceso a su información.
- Laravel es un Framework para el desarrollo de aplicaciones escritas en el Lenguaje PHP que utiliza el MVC, que permite a los desarrolladores Web mejorar la forma de trabajar y también agregar mayor velocidad a la hora de crear una página Web.

## **1.3 REQUISITOS**

### **1.3.1 HARDWARE**

- Procesador de 32 bits (x86) o 64 bits (x64) a 1 gigahercio (GHz) o más.
- Memoria RAM de 1 gigabyte (GB) (32 bits) o memoria RAM de 2 GB (64 bits).
- Espacio disponible en disco rígido de 1 GB (32 bits) o 2 GB (64 bits).
- Sistemas Operativos Compatibles Linux, Windows (7 o Superior) o Mac OSX

### **1.3.2 SOFTWARE**

Se debe tener instalado:

- Apache 2 o WAMP Server (Solamente en Windows) o XAMPP
- PHP 5.5
- Base de Datos Postgres versión 9.1 (sólo en caso que se desee ejecutarlo de forma local)
- Navegador de Internet como Chrome, Firefox, Opera, Internet Explorer 8 o Superior.
- Herramientas de Ofimática tales como OpenOffice o Microsoft Office que permitan el uso de Hojas de Cálculos.

## 1.4 INSTALACIÓN

### 1.4.1 Descarga del Programa

El Programa debe ser descargado desde el sitio web github desde la siguiente dirección: [https://github.com/Lanceconan/ISW\\_PCC](https://github.com/Lanceconan/ISW_PCC) esto mostrara la siguiente ventana.

Ahora usted tiene dos formas de descargar nuestra aplicación una es descargar desde un zip. para ello usted en la misma página debe clickear el botón “Download Zip” como se muestra en la Figura 1:

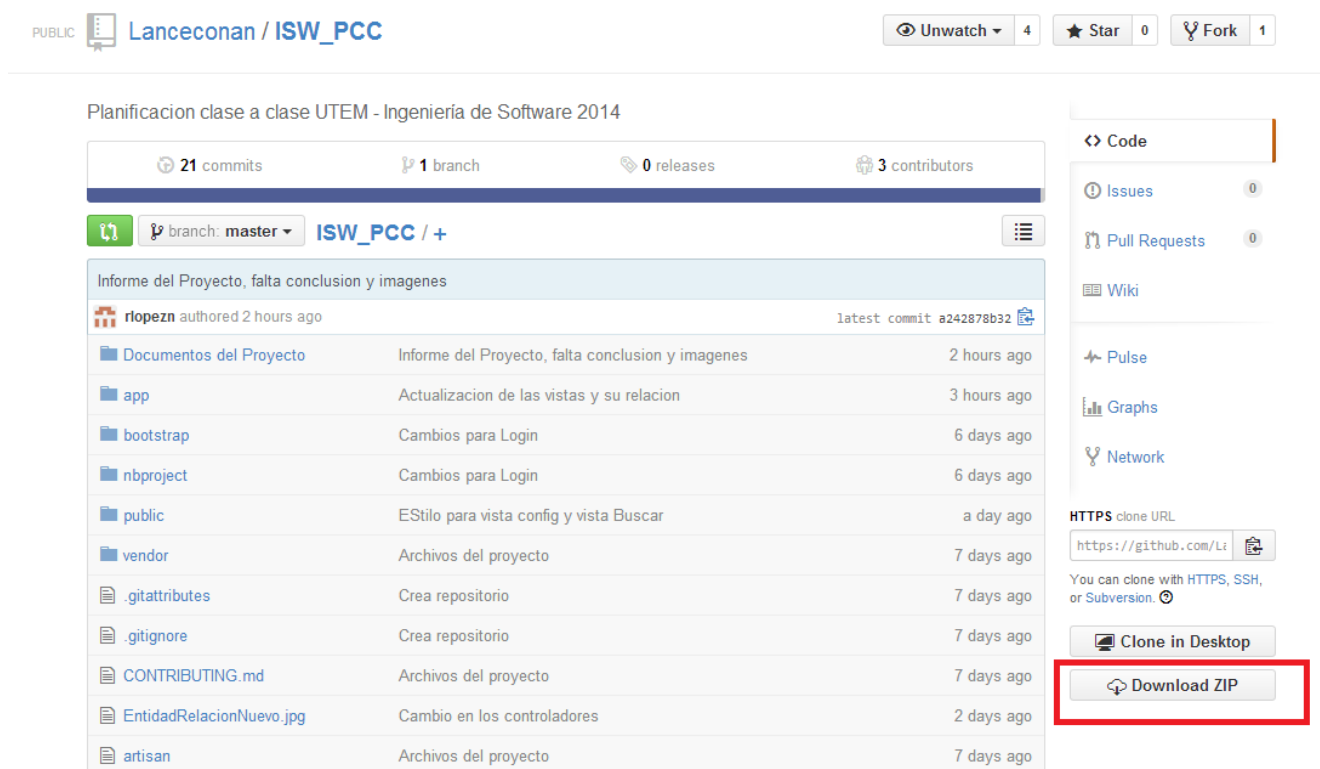


Figura 1: El recuadro rojo indica donde se descarga el proyecto

## 1.4.2 INSTALACIÓN DEL PROGRAMA

Luego que se tiene descargado el proyecto “ISW\_PCC-master” como lo muestra la Figura 2:



Figura 2: Archivo .rar descargado del Repositorio

Se procede a la instalación del programa, descomprimiendolo en la carpeta del directorio donde apache2 guarda las paginas web usualmente es `/var/www` en Sistemas Operativos Linux, en cambio en Windows es por defecto en `c:/wamp/www` si tiene instalado WampServer, la siguiente figura 3, nos muestra la carpeta en donde se debe descomprimir dicho archivo:

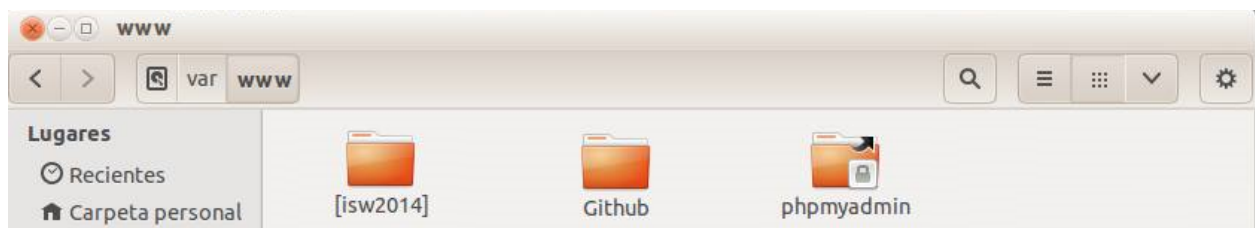


Figura 3: Ruta en donde descomprimir el Proyecto en Linux

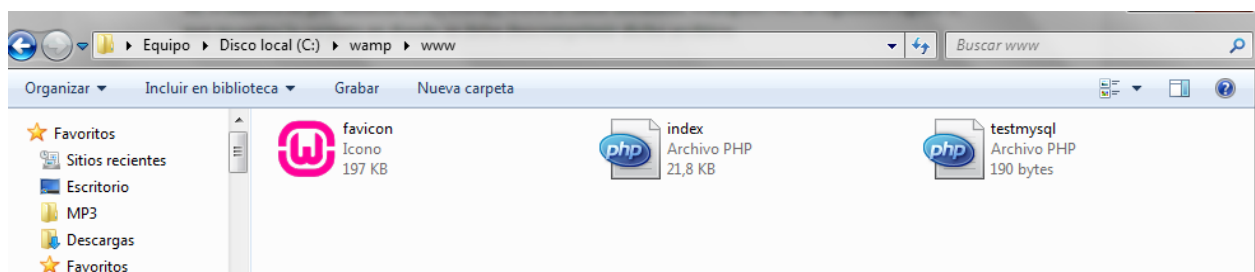


Figura 4: Ruta en donde descomprimir el Proyecto en Windows

## 1.5 PUESTA EN MARCHA DEL PROYECTO

Para que el sitio web funcione correctamente debe configurarse la base de datos. Para poder ver su funcionamiento disponemos de una base de datos montada en un servidor de ejemplo. Dicho servidor viene previamente configurado y con algunos datos de ejemplo.

Para poder comprobar el correcto funcionamiento del sitio web y que los datos por defecto a la conexión del servidor se encuentren adecuadamente correctos ubíquese dentro del directorio

**ISW\_PCC / app / config** / y compruebe si los datos puestos dentro del archivo “database.php” son igual a estos, en caso contrario debe modificarlos según puesto en la Tabla N°1:

```
'default' => 'pgsql', // default database connection name

'pgsql' => array(
    'driver' => 'pgsql',
    'host'   => '146.83.181.4',
    'database' => 'iswdb',
    'username' => 'grupo01',
    'password' => '6fdde6',
    'charset' => 'utf8',
    'prefix' => '',
    'schema' => 'grupo01',
    'port'   => '6432',

),
```

Tabla N°1: Parámetros de Configuración de Base de Datos en el archivo “database.php”



Una vez ya comprobados los datos de conexión a la base de datos, se procede a abrir su navegador y colocar la siguiente url para empezar a usar la aplicación.

[http://localhost/ISW\\_PCC-master/public/](http://localhost/ISW_PCC-master/public/)

Donde se mostrara a continuación la siguiente pagina donde muestra el correcto funcionamiento de la aplicación.

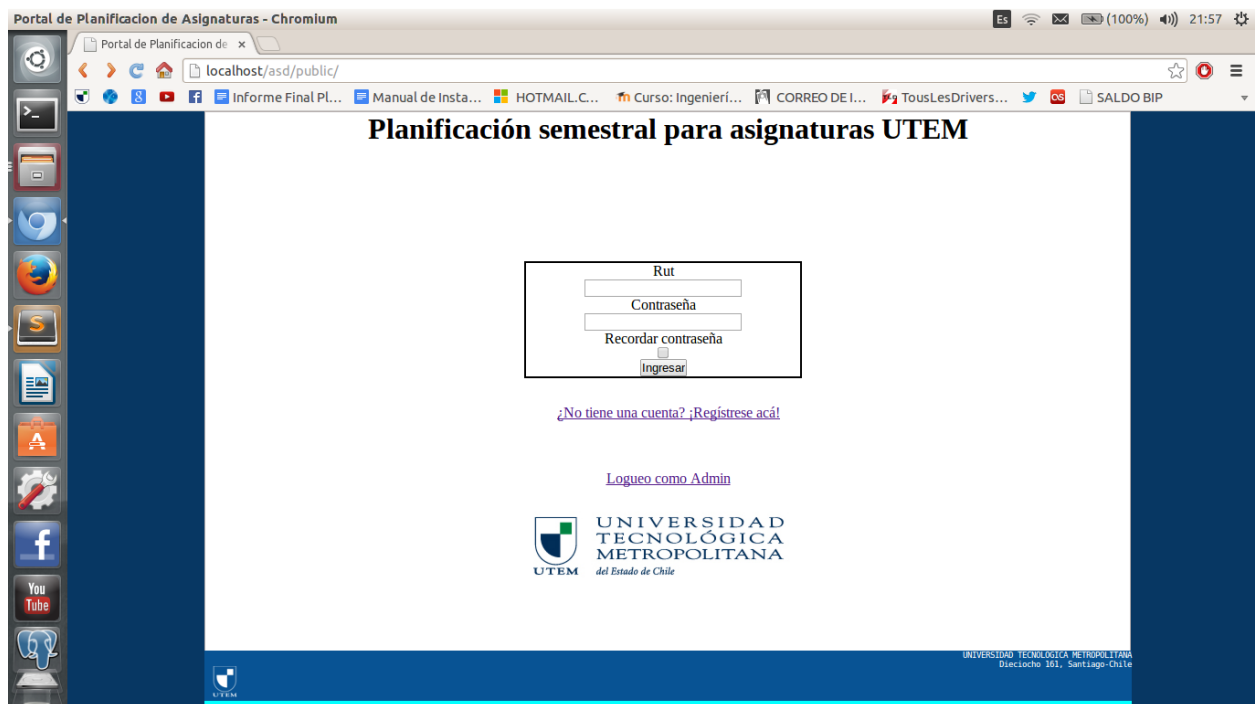


Figura 6: Aplicación funcionando

## 1.6 PREGUNTAS FRECUENTES

1. ¿El Software puede correr en modo local sin necesidad de internet?

a) No puede correr en modo local ya que existe un web service que tiene que incluir conexión a internet para su funcionamiento.

2. ¿En qué Sistemas Operativos puede correr la aplicación?

a) En distribuciones basadas en debian, como ubuntu, etc. Windows xp o superior, y Mac OSX.

3. ¿Es necesario instalar Apache 2?

a) Es necesario solamente en distribuciones linux, en cambio en distribuciones como Windows o Mac OSX existen Aplicaciones como wampServer, Xampp, que traen todo incluido apache, php, que es lo necesario para el funcionamiento de la aplicación.

4. ¿Tengo que tener Office instalado?

a) Para efectos del completo uso de la aplicación es necesario debido al hecho que el usuario tiene el privilegio de descargar su planificación en .xls

## 1.7 ERRORES COMUNES

### 1. Not Found

a) Se debe a que la url no esta bien ingresada, verifique que la url sea la siguiente

[http://localhost/ISW\\_PCC-master/public/](http://localhost/ISW_PCC-master/public/)

### 2. No se ha podido conectar al servidor de base de datos usando la configuración suministrada.

a) Se debe a que hay datos en la parte de la configuración que no es correcto o la conexión a internet esta teniendo problemas, por favor verifique si los datos expuesto a continuación coinciden con su configuración de base de datos y la conexión a internet sea estable.

I). Vaya al directorio **ISW\_PCC** / app / **config** / y compruebe si los datos puestos dentro del archivo “database.php” son igual a estos, en caso contrario debe modificarlos según puesto en la Tabla N°2:

```
'default' => 'pgsql', // default database connection name

'pgsql' => array(
    'driver' => 'pgsql',
    'host' => '146.83.181.4',
    'database' => 'iswdb',
    'username' => 'grupo01',
    'password' => '6fdde6',
    'charset' => 'utf8',
    'prefix' => '',
    'schema' => 'grupo01',
    'port' => '6432',
),
```

Tabla N°2: Configuración predeterminada de Base de Datos

3. No poder borrar,editar en la carpeta www

a) Verifique que la carpeta tenga los permisos correspondiente en el sistema operativo que este utilizando.

1) Linux: `sudo chmod 777 var/www`

2) Windows: Dar permisos de administrador.

#### 4.CONFIGURACIÓN DE UN REPOSITORIO CENTRAL (SERVIDOR)

- Vamos a configurar un repositorio central
- y dos repositorios locales para los desarrolladores que trabajen en él.

##### Instalar GIT

```
root@dev:~# sudo apt-get install git
```

##### Configuración de usuarios

```
root@dev:~# git config --global user.name "Argos"
```

```
root@dev:~# git config --global user.email argos@hd.com
```

##### Instalación de openssh (necesario para transferir datos mediante el protocolo SSH)

```
root@dev:~# apt-get install openssh-server
```

##### Creación del usuario "git" (propietario del repositorio)

```
root@dev:~# adduser git
```

Configuramos el servidor para autenticarnos mediante llaves públicas en vez de hacerlo con contraseñas.

##### Abrimos el archivo "sshd\_config"

```
root@dev:~# nano /etc/ssh/sshd_config
```

##### Agregamos las siguientes líneas:

```
PubkeyAuthentication yes
```

```
AuthorizedKeysFile .ssh/authorized_keys
```

Ahora, creamos las llaves públicas:

Cambiamos al usuario "git" que creamos anteriormente

```
root@dev:~# su git
```

Nos movemos al "home" del usuario

```
git@dev:/home/argos$ cd ~
```

Creamos la carpeta oculta "ssh" para almacenar las llaves públicas

```
git@dev:~$ mkdir .ssh
```

Creamos el archivo "authorized\_keys":

```
git@dev:~$ touch .ssh/authorized_keys
```

Creamos una carpeta para guardar las llaves públicas que nos envíen, así tendremos un mejor control de quienes se están comunicando con nuestro servidor.

```
git@dev:~$ mkdir desarrolladores
```

Finalmente, crearemos un directorio que almacenará nuestro repositorio:

```
git@dev:~$ mkdir dev
```

```
git@dev:~$ cd dev
```

A continuación, inicializamos nuestro primer proyecto, al que llamaremos "cronos"

```
git@dev:~/dev$ mkdir cronos.git
```

```
git@dev:~/dev/cronos.git$ cd cronos.git
```

```
git@dev:~/dev/cronos.git$ git --bare init
```

Hemos inicializado un repositorio vacío que se utilizará como repositorio central. Aquí encontrarán todos los commit que vayan realizando al proyecto, sin embargo este repositorio no se usa para desarrollar.

## CONFIGURACIÓN DE LOS REPOSITARIOS LOCALES

Las siguientes instrucciones deberán seguirse en el ordenador de cada uno de los desarrolladores

El primer paso, será instalar y configurar GIT.

```
root@jose:~# apt-get install git
```

```
root@jose:~# git config --global user.name "Jose"
```

```
root@jose:~# git config --global user.email jose@hd.com
```

A continuación, generamos la llave pública y la enviamos al servidor:

```
root@jose:~# ssh-keygen -t rsa
```

Nos pide una frase clave para usar la llave, esta tiene que ser mayor a 4 dígitos. Luego,

enviamos la llave pública a la carpeta desarrolladores en nuestro servidor.

```
root@jose:~# scp /root/.ssh/id_rsa.pub git@192.168.1.9:desarrolladores/jose_rsa.pub
```

A continuación, en el servidor, agregaremos la llave enviada a `authorized_keys`:

```
git@dev:~$ cat desarrolladores/jose_rsa.pub >> .ssh/authorized_keys
```

Nuevamente, en el ordenador del desarrollador, crearemos un directorio para el proyecto:

```
root@jose:~# mkdir cronos
```

```
root@jose:~# cd cronos/
```

Ahora agregaremos un archivo como prueba

```
root@jose:~/cronos# echo "hola HD :P " >> README
```

Inicializamos GIT

```
root@jose:~/cronos# git init
```

Agregar todos los archivos que se desea enviar al servidor También se puede enviar archivo por archivo o por tipo de archivos.

```
root@jose:~/cronos# git add .
```

Enviamos nuestro primer commit

```
root@jose:~/cronos# git commit -m "Prueba del primer commit"
```

Agregamos la ruta del repositorio central

```
root@jose:~/cronos# git remote add origin git@192.168.1.9:~/dev/cronos.git
```



Enviamos los cambios al repositorio central

```
root@jose:~/cronos# git push origin master
```

Le pedirá que ingrese la frase con la que creo la clave pública

Para que otros desarrolladores puedan recuperar los cambios enviados, se utiliza el comando "pull" (en vez de push):

```
root@jose:~/cronos# git pull origin master
```

Para visualizar el historial de cambios se puede utilizar una GUI como GitWeb o sino, por línea de comandos mediante git log:

```
root@jose:~/cronos# git log --stat
```