



**UNIVERSIDAD RÓMULO GALLEGOS**

**DECANATO DE INVESTIGACIÓN Y EXTENSIÓN**

**MANUAL TÉCNICO**

**SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE LAS  
ACTIVIDADES DE INVESTIGACIÓN Y EXTENSIÓN DE LA  
UNIVERSIDAD RÓMULO GALLEGOS**



## ÍNDICE

ÍNDICE.....	2
INTRODUCCIÓN.....	3
SOBRE EL SISTEMA WEB .....	4
CONFIGURACIÓN GENERAL .....	5
Url's Amigables.....	5
Configurando URL Amigables en Linux .....	5
Configurando URL Amigables en Windows.....	5
REQUISITOS DEL SERVIDOR .....	6
Software.....	6
Hardware. ....	7
INSTALACIÓN DEL SISTEMA .....	7
Instalación en Linux. ....	7
Instalación en Windows.....	8
Configurando Archivo de Conexión.....	8
DISEÑO DEL SISTEMA.....	9
Lenguajes.....	9
Base de Datos. ....	10
Patrón de Diseño.....	11
Técnicas de Documentación en Código. ....	12
RECOMENDACIONES .....	13

## INTRODUCCIÓN

El presente manual está dirigido a todos aquellos usuarios técnicos o profesionales que deseen conocer, los aspectos técnicos del Sistema de Información Web para la Gestión de las Actividades de Investigación y Extensión, para de alguna manera adentrarse y tener un idea sobre cómo fue creado el software, pensando en futuros mantenimientos o expansiones.

Se presentará y detallará sobre los lenguajes de programación utilizados, la base de datos utilizada, el patrón de diseño, los requisitos y aplicaciones del servidor para poder operar, y más detalles sobre la estructura interna del sistema; esperando, como ya se dijo, que sirva de iniciación para futuros desarrolladores.

## **SOBRE EL SISTEMA WEB**

El Sistema de información web para la gestión de las actividades de investigación y extensión de la universidad Rómulo Gallegos es una herramienta diseñada para gestionar dos de las principales tareas que cumple el Decanato de Investigación y Extensión de esta casa de estudios que son: a) la gestión de todo el contenido de investigación y extensión y b) la gestión de los espacios. Entendiéndose la primera como el manejo de toda la información acerca de los centros de investigación, los proyectos de investigación o extensión, las líneas, los investigadores y extensionistas, etc., y la segunda como el manejo y gestión de las actividades y eventos que se realizan tanto en el Auditorio “Hugo Rafael Chávez Frías” como en el Salón de Usos múltiples.

## CONFIGURACIÓN GENERAL

### Url's Amigables

El Sistema de Información trabaja con las Url's amigables, usando el archivo .htaccess. Como su propio nombre lo indica las URL amigables son más fáciles de recordar por los usuarios y también se dice (y hay quienes dicen que no) que son mejores para el posicionamiento en buscadores. Además describen mucho mejor los enlaces dentro del sistema y añade un mayor nivel de profesionalidad al mismo.

### Configurando URL Amigables en Linux

Para configurar nuestro servidor para que acepte las URL amigables basta con introducir los siguientes comandos en la consola:

```
sudo a2enmod rewrite
sudo /etc/init.d/apache2 restart
```

A continuación editamos el archivo /etc/apache2/sites-enabled/000-default y buscamos la línea AllowOverride None y la cambiamos por AllowOverride All.

```
sudo nano /etc/apache2/sites-enabled/000-default
```

Con esto activamos el módulo mod\_rewrite (el que permite reconocer este tipo de URL) de apache2 y reiniciando apache2 aplicamos los cambios.

### Configurando URL Amigables en Windows

En Windows debemos modificar el archivo httpd.conf que dependiendo del servidor que hayamos instalado su ubicación será diferente. En ese archivo buscamos la línea "LoadModule rewrite\_module modules/mod\_rewrite.so" (sin comillas) y si tiene el caracter # es porque está comentado, borramos ese caracter.

Después buscamos esta sección:

```
<Directory />
Options FollowSymLinks ExecCGI Indexes
AllowOverride All
Order deny,allow
Deny from all
Satisfy all
</Directory>
```

Y debemos poner AllowOverride All . Como está en la imagen anterior.

## REQUISITOS DEL SERVIDOR

### Software.

**Sublime Text:** para futuros desarrolladores se recomienda usar el editor de texto Sublime Text en su última versión, por su gran número de bondades y prestaciones que, sin duda, ayudan en el trabajo de la programación.



**Navegador Web:** es un software, aplicación o programa que permite el acceso a la Web, interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser visualizados. Se recomienda utilizar versiones actuales Mozilla Firefox o Google Chrome.



**Servidor Local:** si se desea desarrollar en Linux, es conveniente saber que todas las distribuciones traen consigo instalado el Apache, a diferencia de Windows en donde sí se debe instalar una aplicación como el XAMPP que funja de servidor local.



**PostgreSQL:** el sistema de información web para el Decanato de Investigación y Extensión, para su base de datos utiliza PostgreSQL 9.3. Se recomienda utilizar esta versión.



### **Hardware.**

A nivel de Hardware no se requieren mayores requisitos debido a que el sistema es basado en la web, es decir, que mientras se tenga un equipo que soporte las aplicaciones y/o programas antes mencionados, entonces se podrá desarrollar en el mismo.

## **INSTALACIÓN DEL SISTEMA**

Es necesario contar con el código fuente del sistema, el cual se puede decir que está dentro de la carpeta “investigación”. Teniendo esto, ya podemos recurrir a algunas de estas acciones de acuerdo a nuestro sistema operativo.

### **Instalación en Linux.**

Para instalar o importar el sistema del decanato al equipo lo primero que se debe hacer es copiar la carpeta “investigación” en la ruta /var/www/. Posteriormente se deben dar los debidos permisos a la carpeta recién copiada tipeando la siguiente línea en la consola:

- `Chmod -R 755 /var/www/investigacion`

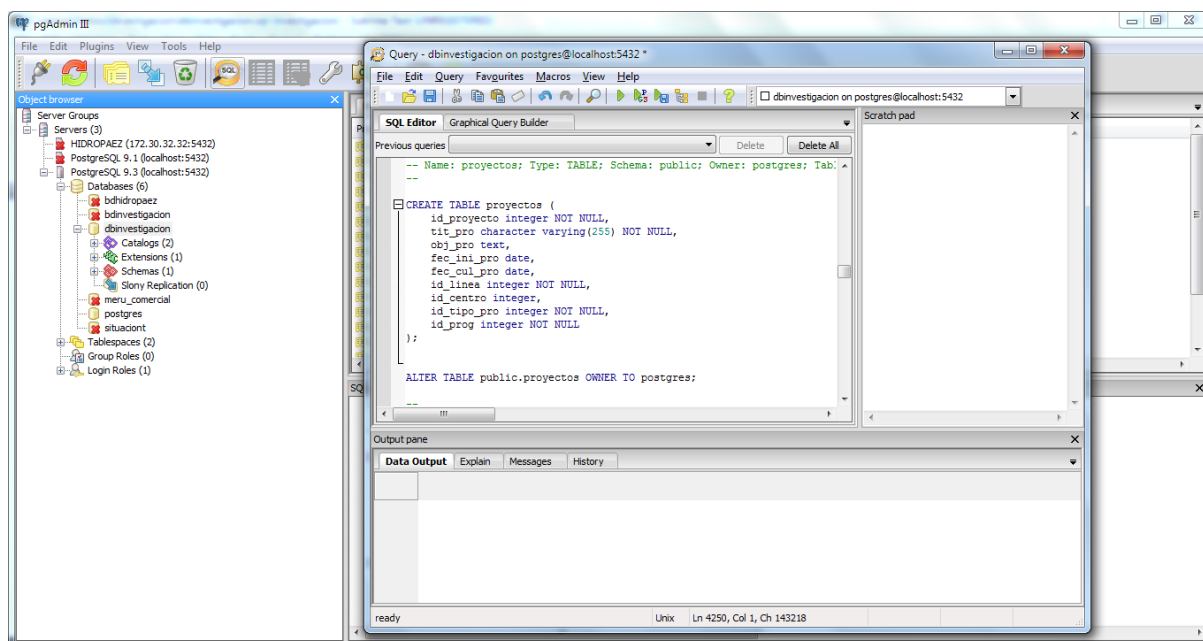
El siguiente paso es importar la base de datos del sistema que se encuentra en la carpeta “BD” ubicada dentro de “investigación”. Si se abre el archivo `bdinvestigacion.sql` se podrá observar que consiste en todo el sql que da a lugar a la estructura de la base de datos, por lo que, una vez que se haya creado la base de datos “dbinvestigación”, se ejecuta el código sql a modo de script para crear la base de datos.

Y listo. Con eso ya debería ser posible ingresar al sistema por nuestro navegador web por la siguiente ruta: `/localhost/investigacion`

### Instalación en Windows

El proceso es exactamente igual. En primer lugar se copia la carpeta del sistema en el servidor local, los nombres de las carpetas de los servidores locales varían según los mismos. Por ejemplo en XAMPP, el proyecto se debe copiar en la carpeta “htdocs”.

Y luego se crea la base de datos en PostgreSQL, usando el archivo “dbinvestigacion.sql”, como se ve a continuación:



### Configurando Archivo de Conexión.

Para completar la correcta instalación del sistema se debe abrir el archivo `db_conexion.php` ubicado en `/investigación/core/` y modificar las variables `$db_user` (con el nombre del usuario en la base de datos), `$db_pass` (con la clave que se tenga en la base de datos) y `$db_host` (si aplica, con la dirección del servidor).



## DISEÑO DEL SISTEMA

### Lenguajes.

El sistema fue creado utilizando los siguientes lenguajes de programación:

**PHP:** es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.



**CSS:** es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML.



**HTML:** hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, entre otros.



**JavaScript:** es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.



**Librerías jQuery:** es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. En el sistema se usaron algunas como jquery.validate (para la validación de los formularios) y jquery.datatables (para el manejo de las tablas dinámicas que se ven en todo el sistema).



## Base de Datos.

Como se dijo anteriormente el Sistema Manejador de Bases de Datos usado en este sistema fue PostgreSQL, en su versión 9.3.

La base de datos del sistema consta de 36 tablas normalizadas hasta la 3era forma normal. En el momento de diseñar la base de datos se utilizó la herramienta de PostgreSQL pgAdmin 3.

El nombre de las tablas de la base de datos siempre hace alusión a la entidad de donde proviene, por lo que no es muy difícil saber qué se guarda en cada tabla.

Table	Owner	Comment
autoridad_cargo	postgres	Tabla para Parametrizar los Cargos de las Autoridades
autoridad_email	postgres	Tabla para guardar los Email de las Autoridades
autoridad_tel	postgres	Tabla para guardar los teléfonos de las autoridades
autoridades	postgres	Tabla para Guardar las Autoridades del Decanato
centros	postgres	Tabla para Guardar los Centros de Investigación
esp_inv	postgres	Tabla para parametrizar las especialidades de los Investigadores
evento_area	postgres	Tabla para parametrizar las Areas académicas
evento_espacio	postgres	Tabla para guardar los espacios
evento_info	postgres	Tabla para Guardar Fechas y Horas de Eventos
evento_publico	postgres	Tabla para parametrizar los tipos de públicos
evento_tipo	postgres	Tabla para parametrizar los tipos de eventos
eventos	postgres	Tabla para Guardar Eventos Aprobados
info_decanato	postgres	Tabla para Guardar Información General del Decanato
info_decanato_j...	postgres	Tabla para Guardar las Imágenes del Carousel de Inicio
info_decanato_jor	postgres	Tabla para guardar la información de las jornadas
info_decanato_h...	postgres	Para Parametrización de las Normas para Usar el Decanato
info_decanato_p...	postgres	Tabla para Guardar los programas del decanato
investigador_email	postgres	Tabla para guardar los correos de los investigadores
investigador_esp	postgres	Tabla para asociar los investigadores con las especialidades
investigador_tel	postgres	Tabla para guardar los teléfonos de los investigadores
investigadores	postgres	Tabla de Investigadores
lineas	postgres	Tabla para Guardar las Lineas de Investigación
modulos	postgres	Tabla de Módulos
movimiento_usuario	postgres	Bitacora de Movimientos por Usuario
movimientos	postgres	Tabla de Movimientos. Bitacora del Sistema
noaprob_causa	postgres	Tabla para parametrizar las Causas de NO aprobación
noticia_img	postgres	Tabla para guardar las imágenes de las Noticias
noticias	postgres	Tabla para Guardar las Noticias Que se Mostrarán en el Inicio
permisos	postgres	Tabla de permisos
proyecto_inv	postgres	Tabla para Guardar los Investigadores cada proyecto
proyecto_tipo	postgres	Tabla para parametrizar los tipos de proyectos
proyectos	postgres	Tabla para Guardar los Proyectos
solic_esp	postgres	Tabla de Solicitudes para el Auditorio
usuario_email	postgres	Tabla para Almacenar los Correos de los Usuarios
usuarios	postgres	Tabla de Usuarios

Además todas las tablas están documentadas completamente, permitiendo determinar la utilidad de cada campo definido en cada una de ellas, con la ayuda del SQL pane de pgAdmin3 como se ve a continuación con la tabla “usuarios”:

```
SQL pane
-- Table: usuarios
-- DROP TABLE usuarios;

CREATE TABLE usuarios
(
    id_usuario serial NOT NULL, -- Identificador del Usuario
    usuario character varying(40) NOT NULL, -- Nombre de Acceso del Usuario
    clave character varying(100) NOT NULL, -- Clave de Acceso al Sistemaa
    nom_usu character varying(40) NOT NULL, -- Nombre de Pila del Usuario
    ape_usu character varying(40) NOT NULL, -- Apellidos del Usuarios
    estado character(1) NOT NULL, -- Estado del Usuario, A->Activo, I->Inactivo
    CONSTRAINT pkusuarios PRIMARY KEY (id_usuario ),
    CONSTRAINT unique_usuario UNIQUE (usuario ),
    CONSTRAINT check_estado CHECK (estado = 'A'::bpchar OR estado = 'I'::bpchar)
);
WITH (
    OIDS=FALSE
);
ALTER TABLE usuarios
    OWNER TO postgres;
COMMENT ON TABLE usuarios
    IS 'Tabla para Almacenar los Usuarios del Sistema';
COMMENT ON COLUMN usuarios.id_usuario IS 'Identificador del Usuario';
COMMENT ON COLUMN usuarios.usuario IS 'Nombre de Acceso del Usuario';
COMMENT ON COLUMN usuarios.clave IS 'Clave de Acceso al Sistemaa';
COMMENT ON COLUMN usuarios.nom_usu IS 'Nombre de Pila del Usuario';
COMMENT ON COLUMN usuarios.ape_usu IS 'Apellidos del Usuarios';
COMMENT ON COLUMN usuarios.estado IS 'Estado del Usuario, A->Activo, I->Inactivo';
```

## Patrón de Diseño

El patrón de diseño utilizado fue el MVC (modelo, vista, controlador) que es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

En el sistema esto se puede ver reflejado en su estructura de directorios que es la siguiente:



Donde:

- DB: contiene el script para la creación de la base de datos.
- CONTROLADORES: almacena los archivos que sirven para relacionar la vista con los modelos, dependiendo de las peticiones de los usuarios.
- MODELOS: archivos que definen las clases y/o objetos, que interactúan con la base de datos.
- VISTAS: que contiene todos los archivos .html que da a lugar a la interfaz del sistema.
- PUBLICO: que almacena todo aquello que es de acceso público, como los archivos css, y los js.
- CORE: que contiene funciones y métodos que son generales para todos los modelos y para todos los controladores, así como también los archivos de conexión a la base de datos.
- .HTACCESS: archivo donde se almacena cada una de las URL amigables, que redirige siempre a un controlador para que haga el trabajo.

## Técnicas de Documentación en Código.

Dentro del código se observará que se documentó cada modelo de manera que, antes de comenzar algún método se especifica para que sirve o qué hace como se refleja en la siguiente imagen:

```
55 if ($id_area!='') {
56
57
58 //Si el evento es para que se apruebe de inmediato
59 if ($est_evento=='A') {
60
61 $cantidadFechas = count($fecha);
62 for ($i=0; $i < $cantidadFechas; $i++) {
63
64     //Verifico que las horas ingresadas sean correctas
65
66     if ($hora_hasta[$i]<=$hora_desde[$i]) {
67         $this->mensaje = "La Hora de Fin, Siempre tiene que ser después que la de Inicio. Revis
68         $this->tipomsg = "danger";
69         $horasBien = false;
70         break;
71     }else{
72
73     }
74
75     //Verifico que No sea una fecha pasada
76     if ($fecha[$i]<$fechaactual) {
77         $this->mensaje = "No puede Agregar Un Evento en el Pasado! Error, Intente de Nuevo";
78         $this->tipomsg = "danger";
79         $fechaPasada = true;
80         break;
81     }else{
82
83     }
84
85     //verificar continuidad de fechas
86     if ($cantidadFechas>1) {
87         if ($i>0) {
88             $diaAnterior = $fecha[$i-1];
```

## RECOMENDACIONES

Se recomienda a futuros desarrolladores seguir respetando el patrón de diseño empleado, manteniendo el orden y la separación de conceptos. Igualmente continuar con las técnicas de documentación para mantener siempre la información de qué es lo que hace cada cosa en el código.