



Manual Técnico

Proyecto: Sistema de Gestión Documental

Universidad Católica del Ecuador

Secretaría General

Autor: Cristhian Recalde

Octubre, 2023



Sistema de Gestión
Documental

MANUAL TÉCNICO

CONTENIDO

Tabla de contenido

CONTENIDO	3
1. Objetivos	4
1.1 Objetivos Específicos.	4
2. Alcance.....	4
3. Requerimientos Técnicos y Hardware	4
3.1 Requerimientos Mínimos de Software.	5
4. Herramientas Utilizadas para el Desarrollo.....	5
• Entorno de Desarrollo Integrado (IDE) - Visual Studio Code	5
• CodeIgniter - Framework de Desarrollo de PHP	5
5. Instalación.....	5
6. Configuración	6
7. Contingencias y soluciones.....	7
8. Tablas.....	7
9. Codificación Reportes Académicos.....	9
10. Codificación Grafica Estadística.....	12
11. Codificación Subir Datos.....	18
12. Codificación Ficheros.....	27

1. Objetivos

Este sistema fue desarrollado con el objetivo de simplificar y mejorar significativamente la forma en que las secretarías gestionan la información esencial para generar reportes estadísticos de los estudiantes.

1.1 Objetivos Específicos.

- Gestión eficiente de datos estadísticos de estudiantes relacionados con matriculados, graduados.
- Generar informes detallados y precisos sobre matriculados y graduados, facilitando así el análisis y la toma de decisiones informadas.
- Validar y mantener la precisión de los datos almacenados, minimizando así los errores y garantizando la confiabilidad de la información.
- Automatizar tareas administrativas relacionadas con la gestión documental

2. Alcance.

Este documento está dirigido a: [Msg. José Luis Ibarra](#)

Conocimientos básicos en:

- Frontend:
 - HTML
 - Css3, Bootstrap
 - JavaScript, JQuery, HighChart, DataTable, FontAwesome
- Backend:
 - PHP
 - Framework: CodeIgniter
- Base de Datos:
 - MySQL
- Control de Versiones:
 - Git, GitHub

3. Requerimientos Técnicos y Hardware

- Memoria RAM mínima: 4gb
- Conexión a Internet
- PHP 7.x o superior
- CodeIgniter v4.x
- MySQL 8.x

3.1 Requerimientos Mínimos de Software.

Privilegios de Administrador: **Sí**

Sistema Operativo: **Windows 10 / Ubuntu**

4. Herramientas Utilizadas para el Desarrollo.

- **Entorno de Desarrollo Integrado (IDE) - Visual Studio Code**
Visual Studio Code fue elegido por su interfaz intuitiva y extensibilidad. Su soporte integrado para **Git** simplificó el control de versiones
- **CodeIgniter - Framework de Desarrollo de PHP**
CodeIgniter fue elegido por su facilidad de uso y su capacidad para simplificar el desarrollo web en **PHP**. Su estructura **MVC** ayudó a organizar el código de manera efectiva

5. Instalación

1. *Instalar XAMP y usar el servidor Apache y MySQL*
2. *Importar la Base de Datos con el nombre `gestion_documental`*
 - *Script de la Base de Datos: [link](#)*
3. *En la ruta de XAMPP `C:\XAMPP\htdocs`, crear el directorio `SistemaGestionDocumetal` e importar el proyecto o clonar el repositorio de GitHub en la misma ruta*
 - Repositorio en Línea: [GitHub](#)
 - Nota: El repositorio cuenta con 385 versiones desde su creación

6. Configuración

- *Base url:*

```
public string $baseURL = 'http://localhost:8080/';
```

- *Data Base*

```
public array $default = [  
    'DSN'      => '',  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => '',  
    'database' => 'gestion_documental',  
    'DBDriver' => 'MySQLi',  
    'DBPrefix' => '',  
    'pConnect' => false,  
    'DBDebug'  => (ENVIRONMENT !==  
    'production'),  
    'charset'  => 'utf8',  
    'DBCollat' => 'utf8_general_ci',  
    'swapPre'  => '',  
    'encrypt'  => false,  
    'compress' => false,  
    'strictOn' => false,  
    'failover' => [],  
    'port'     => 3306,  
];
```

- *Environment:*

```
CI_ENVIRONMENT = development
```

7. Contingencias y soluciones.

1. *En caso de haber errores se puede restablecer importando la Base de datos, validada desde 1976 hasta el primer periodo de 2022; tomar en cuenta que en caso de restaurar las inserciones en la misma se perderán*
 - a. *Script de la Base de Datos: [link](#)*
2. *En torno a los ficheros en caso de haber algún restablecimiento, cuenta con RARs de respaldo en caso de restaurar un punto de guardado, Calendarios Académicos y Reglamento general de estudiantes cuentan con dicho sistema de recuperación, tomar en cuenta que en caso de restaurar las inserciones en la misma se perderán*
 - a. *Respaldo Calendarios Académicos: [link](#)*
 - b. *Respaldo Reglamento General Estudiantes: [link](#)*
3. *En caso de un fallo general, restaurar el repositorio clonado de GitHub*
 - a. *Repositorio en Línea: [GitHub](#)*

8. Tablas

1. *tbl_estadistica_matriz es la tabla principal y cuenta con 47 campos*

```
protected $table = 'tbl_estadistica_matriz';
protected $primaryKey = 'ESTM_ID ';

protected $allowedFields = [
    'ESTM_TIPO', 'ESTM_CONDICION', 'ESTM_TIPO_GRADO', 'ESTM_PERIODO',
    'ESTM_CARRERA',
    'ESTM_GENERO_H', 'ESTM_GENERO_M', 'ESTM_ETNIA_MESTIZO_H',
    'ESTM_ETNIA_MESTIZO_M',
    'ESTM_ETNIA_INDIGENA_H', 'ESTM_ETNIA_INDIGENA_M', 'ESTM_ETNIA_AFRO_H',
    'ESTM_ETNIA_AFRO_M',
    'ESTM_ETNIA_MONTUBIO_H', 'ESTM_ETNIA_MONTUBIO_M', 'ESTM_ETNIA_MULATO_H',
    'ESTM_ETNIA_MULATO_M',
    'ESTM_ETNIA_NEGRO_H', 'ESTM_ETNIA_NEGRO_M', 'ESTM_ETNIA_BLANCO_H',
    'ESTM_ETNIA_BLANCO_M',
    'ESTM_NACIONALIDAD_EC_H', 'ESTM_NACIONALIDAD_EC_M', 'ESTM_NACIONALIDAD_COL_H',
    'ESTM_NACIONALIDAD_COL_M',
    'ESTM_NACIONALIDAD_ESP_H', 'ESTM_NACIONALIDAD_ESP_M',
    'ESTM_NACIONALIDAD_FRA_H', 'ESTM_NACIONALIDAD_FRA_M',
    'ESTM_NACIONALIDAD_USA_H', 'ESTM_NACIONALIDAD_USA_M',
    'ESTM_NACIONALIDAD_PER_H', 'ESTM_NACIONALIDAD_PER_M',
    'ESTM_NACIONALIDAD_RUM_H', 'ESTM_NACIONALIDAD_RUM_M',
    'ESTM_NACIONALIDAD_CUB_H', 'ESTM_NACIONALIDAD_CUB_M',
    'ESTM_NACIONALIDAD_URC_H', 'ESTM_NACIONALIDAD_URC_M',
    'ESTM_NACIONALIDAD_VEN_H', 'ESTM_NACIONALIDAD_VEN_M',
    'ESTM_DISCAPACIDAD_H', 'ESTM_DISCAPACIDAD_M', 'ESTM_TOTAL', 'ESTM_SEDE',
    'ESTM_ESTADO',
];
```

2. *ESTM_TIPO viene condicionada con la tabla tbl_carrera_tipo*

CTIP_ID	CTIP_NOMBRE	CTIP_ESTADO
1	Posgrado	0
2	Grado	0
3	Tecnología	0

3. *ESTM_CONDICION* viene condicionada con la tabla *tbl_estudiante*

EST_ID	EST_NOMBRE	EST_ESTADO
1	Matriculado	1
2	Egresado	1
3	Graduado	1

4. *ESTM_TIPO_GRADO* viene condicionada con la tabla *tbl_modalida_titulacion*

MODT_ID	MODT_NOMBRE	MODT_ESTADO
1	Trabajo de titulación	1
2	Examen complejo	1
3	Trabajo de la unidad de integración curricular	1

5. *ESTM_PERIODO* viene condicionada con la tabla *tbl_periodo*

PER_ID	PER_AÑO	PER_PERIODO	PER_ULTIMO	PER_ESTADO
1	2022	ABRIL 2022 - AGOSTO 2022	0	0
2	2015	OCTUBRE 2015 - FEBRERO 2016	0	0
3	2021	ABRIL 2021 - AGOSTO 2021	0	0
4	2017	ABRIL 2017 - AGOSTO 2017	0	0
5	2018	ABRIL 2018 - AGOSTO 2018	0	0
6	2019	ABRIL 2019 - AGOSTO 2019	0	0

6. *ESTM_CARRERA* viene condicionada con la tabla *tbl_carrera*, en esta tabla se extrae posterior las carreras y las escuelas

CAR_ID	CTIP_ID	CAR_NOMBRE	CAR_CARRERA	CAR_ESCUELA	CAR_PADREESC	CAR_ACTIVA	CAR_CAMPUS	CAR_ESTADO
5	2	COMUNICACION	0	1	5	No	1	0
4	2	ARQUITECTURA	0	1	4	No	1	0
6	2	DISEÑO	0	1	6	No	1	0
7	2	ECAA	0	1	7	Sí	1	0
8	2	ENCI	0	1	8	Sí	1	0
9	2	GESTURH	0	1	9	Sí	1	0
10	2	DERECHO	0	1	10	Sí	1	0
11	2	CIENCIAS SOCIALES Y HUMANÍSTICAS	0	1	11	Sí	1	0
12	2	INGENIERÍA	0	1	12	Sí	1	0
14	2	AGROPECUARIA	1	0	7	No	1	0
15	2	ZOOTECNIA	1	0	7	Sí	1	0

- Para extraer escuelas *CAR_ESCUELA* = 1

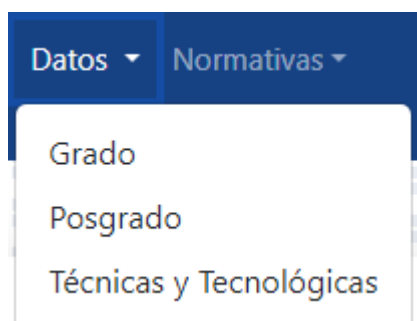
- Para extraer carreras $CAR_CARRERA = 1$

NOTA: ALGUNAS TABLAS CUENTA CON EL CAMPO CAMPUS, PARA DISTINGUIR LA SEDE IBARRA O TULCÁN, DEL MISMO MODO SI ES ACTIVA O NO PARA FILTRAR EN VIGENTE O HISTORICA RESPECTIVAMENTE.

9. Codificación Reportes Académicos

a) El sistema de Gestión Documental funciona en base a 3 condiciones:

- a.1 Grado
- a.2 Posgrado
- a.3 Técnicas y Tecnológicas



b) Para acceder al menú de cada ellas y cambiar la condición:

```
$routes->get('/FiltroEstadisticoGrado',  
'ControladorEstadistico::filtroEstadisticoGrad');
```

c) Cada condición cuenta con 5 filtros

c.1 Escuela, se envía la o las escuelas para mostrar el reporte:

```
$routes->get('/FiltroEstadisticoGradoEscuela/(:any)',  
'ControladorFEEscuela::filtroEstadisticoGradoEscuela/$1');
```

c.2 Carrera, se envía la o las carreras para mostrar el reporte:

```
$routes->get('/FiltroEstadisticoGradoCarrera/(:any)',  
'ControladorFECarrera::filtroEstadisticoGradoCarrera/$1');
```

c.3 Periodo, se envía la o los periodos para mostrar el reporte:



```
$routes->get('/FiltroEstadisticoGradoPeriodo/(:any)',  
'ControladorFEPeriodo::filtroEstadisticoGradoPeriodo/$1');
```

c.4 Fecha, se envía la fecha de inicio y fin para general el reporte:



```
$routes->get('/FiltroEstadisticoGradoBusqueda/(:any)/(:any)',  
'ControladorEstadistico::filtroEstadisticoGradoBusqueda/$1/$2');
```

c.5 General, no se envía nada ya que hace una búsqueda general:



```
$routes->get('/ReporteGeneral',  
'ControladorReportes::reporteGeneral');
```

Resultado:

Filtro	Cantidad
 Escuela	22
 Carrera	180
 Periodo	100
 Fecha	
 General	

d) Cada filtro cuenta con una condición de búsqueda:

d.1 Matriculados

d.2 Graduados

d.3 General



Matriculados



Graduados



General

e) Cada condición de búsqueda cuenta con 3 ofertas académicas en caso de ser el caso

e.1 *Oferta Académica Vigente*

e.2 *Oferta Académica Histórico Campus Ibarra*

e.3 *Oferta Académica Histórico Campus Tulcán*

Resultado

Oferta Académica Vigentes

Mostrar 10 registros

Buscar

Seleccionar	Escuela	Acciones
<input type="checkbox"/>	ECAA	Datos
<input type="checkbox"/>	ENCI	Datos
<input type="checkbox"/>	GESTURH	Datos
<input type="checkbox"/>	DERECHO	Datos
<input type="checkbox"/>	CIENCIAS SOCIALES Y HUMANÍSTICAS	Datos
<input type="checkbox"/>	INGENIERÍA	Datos
<input type="checkbox"/>	ARQUITECTURA DISEÑO Y ARTES	Datos

Mostrando registros del 1 al 7 de un total de 7 registros

Anterior 1 Siguiente

Oferta Académica Histórico

Campus Ibarra

Mostrar 10 registros

Buscar

Seleccionar	Escuela	Acciones
<input type="checkbox"/>	CONSTRUCCIONES CIVILES	Datos
<input type="checkbox"/>	ARQUITECTURA	Datos
<input type="checkbox"/>	COMUNICACION	Datos
<input type="checkbox"/>	DISEÑO	Datos
<input type="checkbox"/>	ECAA	Datos
<input type="checkbox"/>	ENCI	Datos
<input type="checkbox"/>	GESTURH	Datos
<input type="checkbox"/>	DERECHO	Datos
<input type="checkbox"/>	CIENCIAS SOCIALES Y HUMANÍSTICAS	Datos
<input type="checkbox"/>	INGENIERÍA	Datos

Mostrando registros del 1 al 10 de un total de 14 registros

Anterior 1 2 Siguiente

Campus Tulcán

Mostrar 10 registros

Buscar

Seleccionar	Escuela	Acciones
<input type="checkbox"/>	ADMINISTRACIÓN DE EMPRESAS (TULCÁN)	Datos
<input type="checkbox"/>	COMERCIO INTERNACIONAL (TULCÁN)	Datos
<input type="checkbox"/>	CIENCIAS DE LA EDUCACIÓN (TULCÁN)	Datos
<input type="checkbox"/>	SECRETARIADO EJECUTIVO BILINGÜE (TULCÁN)	Datos
<input type="checkbox"/>	INFORMÁTICA (TULCÁN)	Datos
<input type="checkbox"/>	CONSTRUCCIONES CIVILES (TULCÁN)	Datos
<input type="checkbox"/>	ENCI (TULCÁN)	Datos
<input type="checkbox"/>	DISEÑO (TULCÁN)	Datos

Mostrando registros del 1 al 8 de un total de 8 registros

Anterior 1 Siguiente

10. Codificación Grafica Estadística

1. Obtención de Datos:

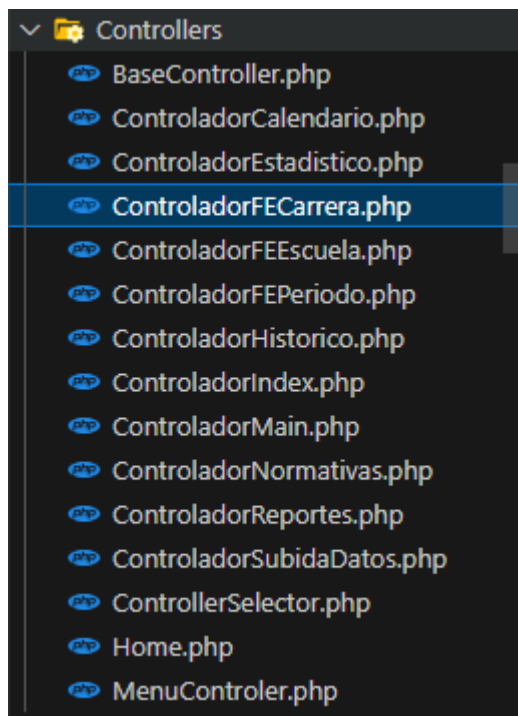
- 1.1 Se utilizan variables PHP (\$tbl_estadistica_matriz, \$tbl_carrera, \$car_id) para obtener datos del servidor. Estos datos son esenciales para generar el gráfico y se almacenan en las variables datos, carrera y carid.

```
var datos = <?php echo json_encode($tbl_estadistica_matriz) ?>;
var carrera = <?php echo json_encode($tbl_carrera) ?>;

//graficar datos acorde a la carrera seleccionada de tbl_estadistica_matriz, donde el id se compara con ESTM_CARRERA
var carid = <?php echo $car_id ?>;
```

2. Controladores

- 2.1 Existen Controladores para cada filtro correctamente separado para mejorar la legibilidad



2.2 Tomaremos de ejemplo el Controlador Carrera, con una búsqueda general

```

public function filtroEstadisticoGradoCarrera($tipo)
{
    try {
        //modelo
        $objEstadMatr = new ModelFEcarreras();
        $objPadre = new ModelFEpadre();

        //car_nombre donde ctip = 2, car_carrera = 1
        $data['tbl_carrera'] = $objEstadMatr
            ->where('CTIP_ID', 2)
            ->where('CAR_CARRERA', 1)
            ->findAll();

        $padre['tbl_escuela'] = $objPadre->findAll();

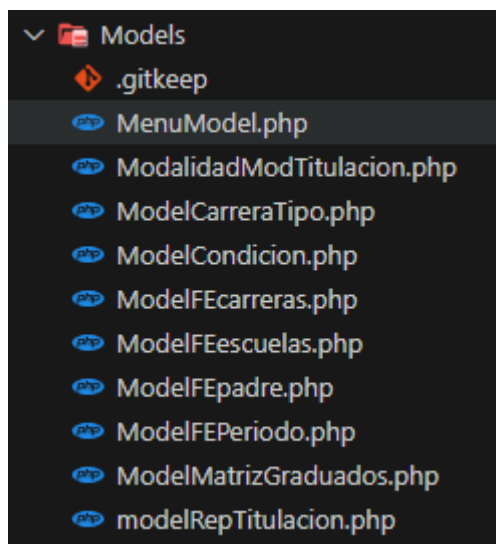
        if ($tipo == "Matriculados") {
            return view('header')
                . view(
                    '/DatosEstadisticos/Grados/busqueda/vista_b_carrera_matr', $data + $padre)
                . view('footer');
        } else if ($tipo == "Graduados") {
            return view('header')
                . view(
                    '/DatosEstadisticos/Grados/busqueda/vista_b_carrera_grad', $data + $padre)
                . view('footer');
        } else if ($tipo == "General") {
            return view('header')
                . view(
                    '/DatosEstadisticos/Grados/busqueda/vista_b_carrera_general', $data + $padre)
                . view('footer');
        }
    } catch (\Exception $e) {
        echo $e->getMessage();
    }
}

```

- Se usan modelos para realizar la búsqueda correspondientemente, y se procesa el tipo de búsqueda para enviar la vista correspondiente en cada solicitud.

3. Modelos

3.1 Cada modelo tiene su relación con su tabla, y tiene sus diferentes métodos, sin embargo, en estos ya están definidos varios procesos que se hace uso en las tablas.



3.2 Tomaremos de ejemplo el modelo FECarreras, FE hace alusión a filtro estadístico.

```
protected $table = 'tbl_carrera';
protected $primaryKey = 'CAR_ID';

protected $allowedFields = ['CTIP_ID', 'CAR_NOMBRE', 'CAR_CARRERA',
'CAR_ESCUELA', 'CAR_PADREESC', 'CAR_ACTIVA', 'CAR_ESTADO'];
```

- Cada modelo tiene su tabla, su llave primaria y los campos almacenados

3.3 Entre las funciones que se usan, la más usada es findAll()

```
public function verModelo()
{
    $carreras = $this->findAll();
    return $carreras;
}
```

- De este modo obtenemos los campos a usar en las graficas

4. Filtrado de Datos:

4.1 Los datos se filtran utilizando la función filter() en JavaScript. Los datos se filtran por el valor de ESTM_CARRERA, ESTM_TIPO y ESTM_CONDICION en el objeto filteredData. Este conjunto de datos filtrado se utiliza para crear el gráfico.

```
// Filtrar datos por ESTM_TIPO y ESTM_CONDICION con la carrera seleccionada, por ESTM_
// TIPO y ESTM_CONDICION
var filteredData = datos.filter(function(dato) {
    return dato.ESTM_CARRERA == carid && (dato.ESTM_TIPO === '2' && (dato.
ESTM_CONDICION === '1' || dato.ESTM_CONDICION === '3'));
});
```

Tipo: 1(Posgrado), 2(Grado), 3(Tecnologías)

Condición: 1(Matriculado), 2(Egresado), 3(Graduado)

5. Cálculo de Totales:

5.1 Se utilizan bucles for para calcular los totales de estudiantes para diferentes categorías (hombres, mujeres y total) en función de la carrera seleccionada. Estos totales se almacenan en las variables totalesH, totalesM y totales.

```
//! Por Años General
{
    // Objeto para asociar carreras con totales
    var carreraTotalMap = {};

    // Recorrer los datos y la carrera
    for (let i = 0; i < filteredData.length; i++) {
// Cambiado a filteredData
        var carrera = filteredData[i].ESTM_CARRERA;
// Obtener la carrera directamente
        // Agregar la carrera si no está en el objeto
        if (!carreraTotalMap.hasOwnProperty(carrera)) {
            carreraTotalMap[carrera] = 0;
// Inicializar el total para esta carrera en 0
        }
        // Acumular el total
        carreraTotalMap[carrera] += parseInt(filteredData[i].ESTM_TOTAL);
// Cambiado a filteredData
    }

    // Obtener las carreras y sus totales
    var carreras = Object.keys(carreraTotalMap).map(Number);
    var totales = carreras.map(function(carrera) {
        return carreraTotalMap[carrera];
    });
}
```

- Misma lógica para totales, totalesH, totalesM

6. Nombre de la Escuela, Carrera, etc.:

6.1 Se extrae el nombre de la escuela del objeto `car_padreesc` y se almacena en la variable `escuela_nombre`. Esto se utiliza posteriormente en el subtítulo del gráfico.

```
//nombre de la escuela
{
    //de datos filtrados extraer car_padreesc
    var car_padreesc = <?php echo json_encode($tbl_carrera) ?>;
    var car_padreesc_id = <?php echo $car_id ?>;
    var car_padreesc_nombre = "";
    for (let i = 0; i < car_padreesc.length; i++) {
        if (car_padreesc[i].CAR_ID == car_padreesc_id) {
            car_padreesc_nombre = car_padreesc[i].CAR_PADREESC;
        }
    }

    //car_padreesc_nombre ahora es un id, buscar el nombre de la escuela en tbl carrera
    var escuela = <?php echo json_encode($tbl_carrera) ?>;
    var escuela_nombre = "";
    for (let i = 0; i < escuela.length; i++) {
        if (escuela[i].CAR_ID == car_padreesc_nombre) {
            escuela_nombre = escuela[i].CAR_NOMBRE;
        }
    }
}
```

7. Configuración del Gráfico:

8. Se configuran diversas opciones del gráfico, como el tipo de gráfico (columnas), el título, el subtítulo, los ejes, las leyendas y otros aspectos visuales. Los datos calculados se utilizan para las series del gráfico.

9. Ajuste del Tamaño del Gráfico:

9.1 Se ajusta dinámicamente la altura del contenedor del gráfico (container) en función del número de carreras para asegurar que el gráfico sea visible y estéticamente agradable.

```
//tam dinamico
    // Calcular el alto deseado en función de la cantidad de carreras
    var altoDeseado = carreras.length * 100;

    //ALTO MINIMO
    if (altoDeseado < 400) {
        altoDeseado = 400;
    }

    //ALTO MAXIMO
    if (altoDeseado > 500) {
        altoDeseado = 500;
    }
```


10. Marca de Agua:

10.1 Se añade una marca de agua (el logotipo de la institución) al gráfico utilizando la función `this.renderer.image()`.

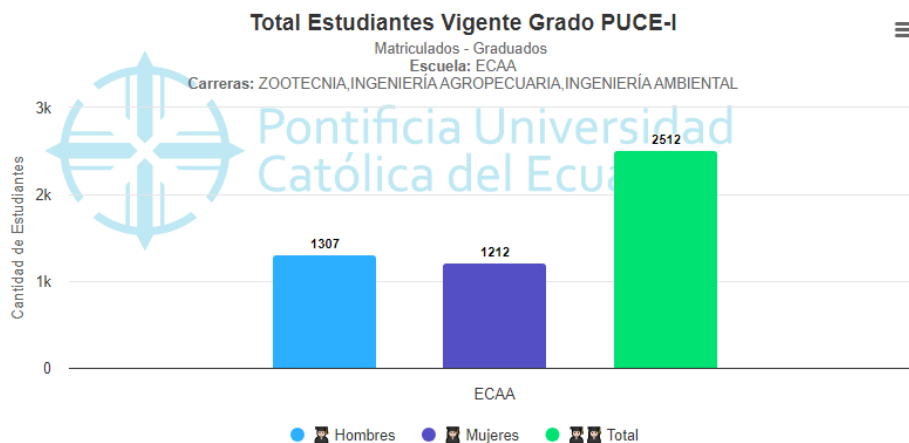
```
chart: {
  type: 'column',
  //marca de agua
  events: {
    load: function() {
      //imagen opaca fondo
      this.renderer.image('<?php echo base_url(
'/public/imgs/logoPucesi.png') ?>')
      .css({
        opacity: 0.35
      })
      .add();
    }
  },
}
```

11. Créditos del Gráfico:

11.1 Se agregan créditos al gráfico que proporcionan un enlace a la página web de la Secretaría General de la institución. Esto se realiza a través de la propiedad `credits` del objeto de configuración del gráfico

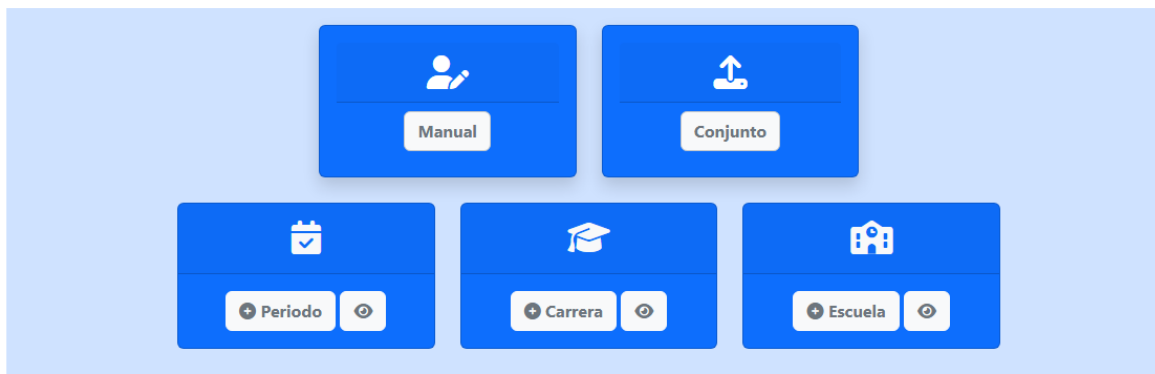
```
credits: {
  enabled: true,
  href: "https://www.pucesi.edu.ec/webs2/",
  text: "Secretaría General PUCE-I",
  style: {
    color: "#666666",
    cursor: "pointer",
    fontSize: "10px"
  },
}
```

12. Resultado



11. Codificación Subir Datos

|| Menú Subir Datos ||



- Para subir los datos al sistema se puede realizar de manera manual y en conjunto, en caso de requerir crear periodos, carreras y escuelas se las puede agregar con el fin que el sistema sea escalable

1. Subida Manual



- Al ser un formulario se adapta a la tbl_estadistica_matriz y se obtiene los campos un datalist para agilizar la búsqueda que se llena dinámicamente con sus respectivas tablas.

- Campo Tipo: tbl_carrera_tipo
- Campo Condición: tbl_estudiante
- Campo Modalidad Titulación: tbl_modalida_titulacion
- Campo Periodo: tbl_periodo, además se puede crear periodos en el mismo
- Campo Carrera: tbl_carrera, además se puede crear carreras en el mismo
- Campo Sede: Se selecciona a la sede que hacemos referencia, Ibarra o Tulcán.
- Campos Género, en estos campos se llenan para Masculino y femenino, sin embargo, también permite llenar dinámicamente según:
 - Etnia
 - Nacionalidad
 - Discapacidad
- Se envía por el método Post al controladorSubida

```

public function enviarManualmente()
{
    try {

        //matriz
        $objEstadMatr = new ModelMatrizGraduados();
        //carrera tipo
        $objCarreraTipo = new ModelCarreraTipo();
        //condicion
        $objCondicion = new ModelCondicion();
        //modalidad
        $objModalidad = new ModalidadModTitulacion();
        //periodo
        $objPeriodo = new ModelFEPeriodo();
        //carrera
        $objCarrera = new ModelFECarreras();

        //obtener datos
        $id_carrera_tipo = $this->request->getPost('id_carrera_tipo');
        $id_condicion = $this->request->getPost('id_condicion');
        $id_tipo_grado = $this->request->getPost('id_tipo_grado');
        $id_periodo = $this->request->getPost('id_periodo');
        $id_carrera = $this->request->getPost('id_carrera');
        $sede = $this->request->getPost('sede');

        $cantmasgen = $this->request->getPost('cantmasgen');
        $cantfemgen = $this->request->getPost('cantfemgen');
        $totalgen = $this->request->getPost('totalgen');
    }
}

```

- Se envían los modelos y se obtienen los datos

```
//procesar datos dinamicamente
//etnia
$indexetnia = $this->request->getPost('indexetnia');
for ($i = 0; $i < $indexetnia; $i++) {
    $etnia[$i] = $this->request->getPost('etnia_' . $i);
    $cantmasetnia[$i] = $this->request->getPost('cantmasgenetnia_' .
    $i);
    $cantfemetnia[$i] = $this->request->getPost('cantfemgenetnia_' .
    $i);
}

//nacionalidad
$indexnacionalidad = $this->request->getPost('indexnacionalidad');
for ($i = 0; $i < $indexnacionalidad; $i++) {
    $nacionalidad[$i] = $this->request->getPost('nacionalidad_' . $i
    );
    $cantmasnacionalidad[$i] = $this->request->getPost(
    'cantmasgennacionalidad_' . $i);
    $cantfemnacionalidad[$i] = $this->request->getPost(
    'cantfemgennacionalidad_' . $i);
}
```

- En caso de ser generados dinámicamente se obtienen y se recorre a través de un bucle for

```
/**genero
$discgenmas = $this->request->getPost('discgenmas');
$discgenfem = $this->request->getPost('discgenfem');

//CLASIFICAR obtener los ids
$id_carrera_tipop = $objCarreraTipo->obtenerId($id_carrera_tipo);
$id_condicionp = $objCondicion->obtenerId($id_condicion);
$id_tipo_gradop = $objModalidad->obtenerId($id_tipo_grado);
$id_periodop = $objPeriodo->obtenerId($id_periodo);
$id_carrerap = $objCarrera->obtenerId($id_carrera);

//preprocesamiento de datos para enviar a la base de datos
$ESTM_TIPO = $id_carrera_tipop['CTIP_ID'];
$ESTM_CONDICION = $id_condicionp['EST_ID'];
$ESTM_TIPO_GRADO = $id_tipo_gradop['MODT_ID'];
$ESTM_PERIODO = $id_periodop['PER_ID'];
$ESTM_CARRERA = $id_carrerap['CAR_ID'];
$ESTM_GENERO_H = $cantmasgen;
$ESTM_GENERO_M = $cantfemgen;
```

- Una vez procesados se prepara para enviar la primera parte de los datos, las 5 primeras columnas

```
//encherado de variables
$ESTM_ETNIA_MESTIZO_H = 0;
$ESTM_ETNIA_MESTIZO_M = 0;
$ESTM_ETNIA_INDIGENA_H = 0;
$ESTM_ETNIA_INDIGENA_M = 0;
$ESTM_ETNIA_AFRO_H = 0;
$ESTM_ETNIA_AFRO_M = 0;
$ESTM_ETNIA_MONTUBIO_H = 0;
$ESTM_ETNIA_MONTUBIO_M = 0;
$ESTM_ETNIA_MULATO_H = 0;
$ESTM_ETNIA_MULATO_M = 0;
$ESTM_ETNIA_NEGRO_H = 0;
$ESTM_ETNIA_NEGRO_M = 0;
$ESTM_ETNIA_BLANCO_H = 0;
$ESTM_ETNIA_BLANCO_M = 0;
$ESTM_NACIONALIDAD_EC_H = 0;
$ESTM_NACIONALIDAD_EC_M = 0;
$ESTM_NACIONALIDAD_COL_H = 0;
$ESTM_NACIONALIDAD_COL_M = 0;
$ESTM_NACIONALIDAD_ESP_H = 0;
$ESTM_NACIONALIDAD_ESP_M = 0;
$ESTM_NACIONALIDAD_FRA_H = 0;
$ESTM_NACIONALIDAD_FRA_M = 0;
$ESTM_NACIONALIDAD_USA_H = 0;
$ESTM_NACIONALIDAD_USA_M = 0;
$ESTM_NACIONALIDAD_PER_H = 0;
$ESTM_NACIONALIDAD_PER_M = 0;
$ESTM_NACIONALIDAD_RUM_H = 0;
$ESTM_NACIONALIDAD_RUM_M = 0;
$ESTM_NACIONALIDAD_CUB_H = 0;
$ESTM_NACIONALIDAD_CUB_M = 0;
$ESTM_NACIONALIDAD_URC_H = 0;
$ESTM_NACIONALIDAD_URC_M = 0;
$ESTM_NACIONALIDAD_VEN_H = 0;
$ESTM_NACIONALIDAD_VEN_M = 0;
$ESTM_DISCAPACIDAD_H = 0;
$ESTM_DISCAPACIDAD_M = 0;
```

- La segunda parte a enviar primero se ensera, ya que la mayoría de estos campos tienen un valor de 0

```
if ($indexetnia != 0) {
  for ($i = 0; $i < $indexetnia; $i++) {
    if ($etnia[$i] == "MESTIZO") {
      $ESTM_ETNIA_MESTIZO_H = $cantmasetnia[$i];
      $ESTM_ETNIA_MESTIZO_M = $cantfemetnia[$i];
    } else if ($etnia[$i] == "INDIGENA") {
      $ESTM_ETNIA_INDIGENA_H = $cantmasetnia[$i];
      $ESTM_ETNIA_INDIGENA_M = $cantfemetnia[$i];
    } else if ($etnia[$i] == "Afroecuatoriano") {
      $ESTM_ETNIA_AFRO_H = $cantmasetnia[$i];
      $ESTM_ETNIA_AFRO_M = $cantfemetnia[$i];
    } else if ($etnia[$i] == "Montubio") {
      $ESTM_ETNIA_MONTUBIO_H = $cantmasetnia[$i];
      $ESTM_ETNIA_MONTUBIO_M = $cantfemetnia[$i];
    } else if ($etnia[$i] == "Mulato") {
      $ESTM_ETNIA_MULATO_H = $cantmasetnia[$i];
      $ESTM_ETNIA_MULATO_M = $cantfemetnia[$i];
    } else if ($etnia[$i] == "Negro") {
      $ESTM_ETNIA_NEGRO_H = $cantmasetnia[$i];
      $ESTM_ETNIA_NEGRO_M = $cantfemetnia[$i];
    } else if ($etnia[$i] == "Blanco") {
      $ESTM_ETNIA_BLANCO_H = $cantmasetnia[$i];
      $ESTM_ETNIA_BLANCO_M = $cantfemetnia[$i];
    }
  }
}

if ($indexnacionalidad != 0) {
  for ($i = 0; $i < $indexnacionalidad; $i++) {
    if ($nacionalidad[$i] == "Ecuatoriana") {
      $ESTM_NACIONALIDAD_EC_H = $cantmasnacionalidad[$i];
      $ESTM_NACIONALIDAD_EC_M = $cantfemnacionalidad[$i];
    } else if ($nacionalidad[$i] == "Colombiana") {
      $ESTM_NACIONALIDAD_COL_H = $cantmasnacionalidad[$i];
      $ESTM_NACIONALIDAD_COL_M = $cantfemnacionalidad[$i];
    } else if ($nacionalidad[$i] == "Española") {
      $ESTM_NACIONALIDAD_ESP_H = $cantmasnacionalidad[$i];
      $ESTM_NACIONALIDAD_ESP_M = $cantfemnacionalidad[$i];
    } else if ($nacionalidad[$i] == "Francesa") {
      $ESTM_NACIONALIDAD_FRA_H = $cantmasnacionalidad[$i];
      $ESTM_NACIONALIDAD_FRA_M = $cantfemnacionalidad[$i];
    } else if ($nacionalidad[$i] == "Estadounidense") {
      $ESTM_NACIONALIDAD_USA_H = $cantmasnacionalidad[$i];
      $ESTM_NACIONALIDAD_USA_M = $cantfemnacionalidad[$i];
    } else if ($nacionalidad[$i] == "Peruana") {
      $ESTM_NACIONALIDAD_PER_H = $cantmasnacionalidad[$i];
      $ESTM_NACIONALIDAD_PER_M = $cantfemnacionalidad[$i];
    } else if ($nacionalidad[$i] == "Rumana") {
      $ESTM_NACIONALIDAD_RUM_H = $cantmasnacionalidad[$i];
      $ESTM_NACIONALIDAD_RUM_M = $cantfemnacionalidad[$i];
    } else if ($nacionalidad[$i] == "Cubana") {
      $ESTM_NACIONALIDAD_CUB_H = $cantmasnacionalidad[$i];
      $ESTM_NACIONALIDAD_CUB_M = $cantfemnacionalidad[$i];
    } else if ($nacionalidad[$i] == "Ucraniana") {
      $ESTM_NACIONALIDAD_URC_H = $cantmasnacionalidad[$i];
      $ESTM_NACIONALIDAD_URC_M = $cantfemnacionalidad[$i];
    } else if ($nacionalidad[$i] == "Venezolano") {
      $ESTM_NACIONALIDAD_VEN_H = $cantmasnacionalidad[$i];
      $ESTM_NACIONALIDAD_VEN_M = $cantfemnacionalidad[$i];
    }
  }
}
```

- En caso de tener alguno de estos datos se procesa y se asigna el valor acorde al id cambiando el valor enserado

```
//ENVIO A LA BD
$datosInsertar = [
    'ESTM_ID' => null,
    'ESTM_TIPO' => $ESTM_TIPO,
    'ESTM_CONDICION' => $ESTM_CONDICION,
    'ESTM_TIPO_GRADO' => $ESTM_TIPO_GRADO,
    'ESTM_PERIODO' => $ESTM_PERIODO,
    'ESTM_CARRERA' => $ESTM_CARRERA,
    'ESTM_GENERO_H' => $ESTM_GENERO_H,
    'ESTM_GENERO_M' => $ESTM_GENERO_M,
    'ESTM_ETNIA_MESTIZO_H' => $ESTM_ETNIA_MESTIZO_H,
    'ESTM_ETNIA_MESTIZO_M' => $ESTM_ETNIA_MESTIZO_M,
    'ESTM_ETNIA_INDIGENA_H' => $ESTM_ETNIA_INDIGENA_H,
    'ESTM_ETNIA_INDIGENA_M' => $ESTM_ETNIA_INDIGENA_M,
    'ESTM_ETNIA_AFRO_H' => $ESTM_ETNIA_AFRO_H,
    'ESTM_ETNIA_AFRO_M' => $ESTM_ETNIA_AFRO_M,
    'ESTM_ETNIA_MONTUBIO_H' => $ESTM_ETNIA_MONTUBIO_H,
    'ESTM_ETNIA_MONTUBIO_M' => $ESTM_ETNIA_MONTUBIO_M,
    'ESTM_ETNIA_MULATO_H' => $ESTM_ETNIA_MULATO_H,
    'ESTM_ETNIA_MULATO_M' => $ESTM_ETNIA_MULATO_M,
    'ESTM_ETNIA_NEGRO_H' => $ESTM_ETNIA_NEGRO_H,
    'ESTM_ETNIA_NEGRO_M' => $ESTM_ETNIA_NEGRO_M,
    'ESTM_ETNIA_BLANCO_H' => $ESTM_ETNIA_BLANCO_H,
    'ESTM_ETNIA_BLANCO_M' => $ESTM_ETNIA_BLANCO_M,
    'ESTM_NACIONALIDAD_EC_H' => $ESTM_NACIONALIDAD_EC_H,
    'ESTM_NACIONALIDAD_EC_M' => $ESTM_NACIONALIDAD_EC_M,
    'ESTM_NACIONALIDAD_COL_H' => $ESTM_NACIONALIDAD_COL_H,
    'ESTM_NACIONALIDAD_COL_M' => $ESTM_NACIONALIDAD_COL_M,
    'ESTM_NACIONALIDAD_ESP_H' => $ESTM_NACIONALIDAD_ESP_H,
    'ESTM_NACIONALIDAD_ESP_M' => $ESTM_NACIONALIDAD_ESP_M,
    'ESTM_NACIONALIDAD_FRA_H' => $ESTM_NACIONALIDAD_FRA_H,
    'ESTM_NACIONALIDAD_FRA_M' => $ESTM_NACIONALIDAD_FRA_M,
    'ESTM_NACIONALIDAD_USA_H' => $ESTM_NACIONALIDAD_USA_H,
    'ESTM_NACIONALIDAD_USA_M' => $ESTM_NACIONALIDAD_USA_M,
    'ESTM_NACIONALIDAD_PER_H' => $ESTM_NACIONALIDAD_PER_H,
    'ESTM_NACIONALIDAD_PER_M' => $ESTM_NACIONALIDAD_PER_M,
    'ESTM_NACIONALIDAD_PER_M' => $ESTM_NACIONALIDAD_PER_M,
    'ESTM_NACIONALIDAD_RUM_H' => $ESTM_NACIONALIDAD_RUM_H,
    'ESTM_NACIONALIDAD_RUM_M' => $ESTM_NACIONALIDAD_RUM_M,
    'ESTM_NACIONALIDAD_CUB_H' => $ESTM_NACIONALIDAD_CUB_H,
    'ESTM_NACIONALIDAD_CUB_M' => $ESTM_NACIONALIDAD_CUB_M,
    'ESTM_NACIONALIDAD_URC_H' => $ESTM_NACIONALIDAD_URC_H,
    'ESTM_NACIONALIDAD_URC_M' => $ESTM_NACIONALIDAD_URC_M,
    'ESTM_NACIONALIDAD_VEN_H' => $ESTM_NACIONALIDAD_VEN_H,
    'ESTM_NACIONALIDAD_VEN_M' => $ESTM_NACIONALIDAD_VEN_M,
    'ESTM_DISCAPACIDAD_H' => $ESTM_DISCAPACIDAD_H,
    'ESTM_DISCAPACIDAD_M' => $ESTM_DISCAPACIDAD_M,
    'ESTM_TOTAL' => $ESTM_TOTAL,
    'ESTM_SEDE' => $ESTM_SEDE,
    'ESTM_ESTADO' => $ESTM_ESTADO
];
```

- Se crea un array con los datos para enviar a la BD

```
$objEstadMatr->insertar($datosInsertar);

echo "datos insertados correctamente";

//redireccionar
return redirect()->to(base_url() .
```

- El modelo de la matriz toma el array insertándolo y redireccionando correspondientemente

2. Subida por conjunto de datos CSV

- Se hace uso de una plantilla donde ya tiene los campos llenos a través de una lista, en la cual se llena por filas, posterior se debe extraer todos los campos con los datos llenos y pegarlos en el modelo CSV.
- Una vez subido dicho modelo CSV, con el método POST extraemos los datos y la lógica es similar a enviar manualmente


```
//matriz
$objEstadMatr = new ModelMatrizGraduados();
//carrera tipo
$objCarreraTipo = new ModelCarreraTipo();
//condicion
$objCondicion = new ModelCondicion();
//modalidad
$objModalidad = new ModalidadModTitulacion();
//periodo
$objPeriodo = new ModelFEPeriodo();
//carrera
$objCarrera = new ModelFEcarreras();

$data['tbl_estadistica_matriz'] = $objEstadMatr->verModelo();
$data['tbl_carrera_tipo'] = $objCarreraTipo->verModelo();
$data['tbl_estudiante'] = $objCondicion->verModelo();
$data['tbl_modalida_titulacion'] = $objModalidad->verModelo();
$data['tbl_periodo'] = $objPeriodo->verModelo();
$data['tbl_carrera'] = $objCarrera->verModelo();
```

- Se envían los modelos relacionados a la tabla tbl_estadistica_matriz

```
//procesamiento del csv
$archivo = $this->request->getFile('archivo');
$file = fopen($archivo, 'r');
if ($file) {
    // Inicializar el array para almacenar los datos
    $csvData = [];

    // Leer el archivo línea por línea y procesar los datos
    while (($line = fgetcsv($file, 0, ';')) !== false) {
        // Reemplazar los valores entre ;; con 0 en la primera columna
        $line[0] = str_replace(';;', ';0;', $line[0]);

        // Celdas vacías colocar 0 en todas las columnas
        foreach ($line as $value) {
            if ($value === "") {
                $value = 0;
            }
        }

        // Agregar la fila procesada al array de datos
        $csvData[] = $line;
    }

    // Cerrar el archivo
    fclose($file);
}
```

- Se procesa el archivo CSV y se extraen los datos

```
// Obtener filas y columnas después de procesar el archivo
$filas = count($csvData);
$columnas = count($csvData[0]);

echo "Filas: " . $filas . " Columnas: " . $columnas . "<br>";

$datos = [];
for ($i = 0; $i < $filas; $i++) {
    for ($j = 0; $j < $columnas; $j++) {
        $datos[$i][$j] = $csvData[$i][$j];
        // echo $datos[$i][$j] . " ";
    }
    // echo "<br>";
}
```

- Se obtiene las filas y columnas creando un array con los datos extraídos

```
//encerrado de variables
{
    $ESTM_ID = null;
    $ESTM_TIPO = 0;
    $ESTM_CONDICION = 0;
    $ESTM_TIPO_GRADO = 0;
    $ESTM_PERIODO = 0;
    $ESTM_CARRERA = 0;
    $ESTM_GENERO_H = 0;
    $ESTM_GENERO_M = 0;
    $ESTM_ETNIA_MESTIZO_H = 0;
    $ESTM_ETNIA_MESTIZO_M = 0;
    $ESTM_ETNIA_INDIGENA_H = 0;
    $ESTM_ETNIA_INDIGENA_M = 0;
    $ESTM_ETNIA_AFRO_H = 0;
    $ESTM_ETNIA_AFRO_M = 0;
    $ESTM_ETNIA_MONTUBIO_H = 0;
    $ESTM_ETNIA_MONTUBIO_M = 0;
    $ESTM_ETNIA_MULATO_H = 0;
    $ESTM_ETNIA_MULATO_M = 0;
    $ESTM_ETNIA_NEGRO_H = 0;
    $ESTM_ETNIA_NEGRO_M = 0;
    $ESTM_ETNIA_BLANCO_H = 0;
    $ESTM_ETNIA_BLANCO_M = 0;
    $ESTM_NACIONALIDAD_EC_H = 0;
    $ESTM_NACIONALIDAD_EC_M = 0;
    $ESTM_NACIONALIDAD_COL_H = 0;
    $ESTM_NACIONALIDAD_COL_M = 0;
    $ESTM_NACIONALIDAD_ESP_H = 0;
    $ESTM_NACIONALIDAD_ESP_M = 0;
    $ESTM_NACIONALIDAD_FRA_H = 0;
    $ESTM_NACIONALIDAD_FRA_M = 0;
    $ESTM_NACIONALIDAD_USA_H = 0;
    $ESTM_NACIONALIDAD_USA_M = 0;
    $ESTM_NACIONALIDAD_PER_H = 0;
    $ESTM_NACIONALIDAD_PER_M = 0;
    $ESTM_NACIONALIDAD_RUM_H = 0;
    $ESTM_NACIONALIDAD_RUM_M = 0;
    $ESTM_NACIONALIDAD_CUB_H = 0;
    $ESTM_NACIONALIDAD_CUB_M = 0;
    $ESTM_NACIONALIDAD_URC_H = 0;
    $ESTM_NACIONALIDAD_URC_M = 0;
    $ESTM_NACIONALIDAD_VEN_H = 0;
    $ESTM_NACIONALIDAD_VEN_M = 0;
    $ESTM_DISCAPACIDAD_H = 0;
    $ESTM_DISCAPACIDAD_M = 0;
    $ESTM_TOTAL = 0;
    $ESTM_SEDE = 0;
    $ESTM_ESTADO = 0;
}
```

- Enceramos las variables de todas las columnas de la tbl_estadistica_matriz


```
//procesamiento de datos por filas
$datosprocesados = [];
echo "<br>";
for ($i = 0; $i < $filas; $i++) {
    //col 0 tipo
    $tipoEstudio = $csvData[$i][0];
    switch ($tipoEstudio) {
        case 'Posgrado':
            $ESTM_TIPO = 1;
            break;
        case 'Grado':
            $ESTM_TIPO = 2;
            break;
        case 'Tecnología':
            $ESTM_TIPO = 3;
            break;
        default:
            $ESTM_TIPO = 2;
            //predeterminado si no coincide con ninguno
            break;
    }
}
```

- Procesamos por filas y extraemos el tipo de la columna 0

```
//col 1 condicion
$condicion = $csvData[$i][1];
switch ($condicion) {
    case 'Matriculado':
        $ESTM_CONDICION = 1;
        break;
    case 'Egresado':
        $ESTM_CONDICION = 2;
        break;
    case 'Graduado':
        $ESTM_CONDICION = 3;
        break;
    default:
        $ESTM_CONDICION = 1;
        break;
}
```

- Procesamos por filas y extraemos la condición de la columna 1

```
//COL 3 periodo
$periodo = $csvData[$i][3];
//buscar el id del periodo de cada fila en tbl
$id_periodop = $objPeriodo->obtenerId($periodo);
$ESTM_PERIODO = $id_periodop['PER_ID'];

//col 4 carrera
$carrera = $csvData[$i][4];
//buscar el id de la carrera de cada fila en tbl
$id_carrerap = $objCarrera->obtenerId($carrera);
$ESTM_CARRERA = $id_carrerap['CAR_ID'];
```

- Para la columna 3 y 4 extraemos el periodo y la carrera pero hacemos uso de un modelo que hace la búsqueda por las tablas

```
// Añadir los datos procesados al array $datosprocesados
$datosprocesados[$i] = [
    'ESTM_ID' => $ESTM_ID,
    'ESTM_TIPO' => $ESTM_TIPO,
    'ESTM_CONDICION' => $ESTM_CONDICION,
    'ESTM_TIPO_GRADO' => $ESTM_TIPO_GRADO,
    'ESTM_PERIODO' => $ESTM_PERIODO,
    'ESTM_CARRERA' => $ESTM_CARRERA,
    'ESTM_GENERO_H' => $csvData[$i][5],
    'ESTM_GENERO_M' => $csvData[$i][6],
    'ESTM_ETNIA_MESTIZO_H' => $csvData[$i][7],
    'ESTM_ETNIA_MESTIZO_M' => $csvData[$i][8],
    'ESTM_ETNIA_INDIGENA_H' => $csvData[$i][9],
    'ESTM_ETNIA_INDIGENA_M' => $csvData[$i][10],
    'ESTM_ETNIA_AFRO_H' => $csvData[$i][11],
    'ESTM_ETNIA_AFRO_M' => $csvData[$i][12],
    'ESTM_ETNIA_MONTUBIO_H' => $csvData[$i][13],
    'ESTM_ETNIA_MONTUBIO_M' => $csvData[$i][14],
    'ESTM_ETNIA_MULATO_H' => $csvData[$i][15],
    'ESTM_ETNIA_MULATO_M' => $csvData[$i][16],
    'ESTM_ETNIA_NEGRO_H' => $csvData[$i][17],
    'ESTM_ETNIA_NEGRO_M' => $csvData[$i][18],
    'ESTM_ETNIA_BLANCO_H' => $csvData[$i][19],
    'ESTM_ETNIA_BLANCO_M' => $csvData[$i][20],
    'ESTM_NACIONALIDAD_EC_H' => $csvData[$i][21],
    'ESTM_NACIONALIDAD_EC_M' => $csvData[$i][22],
    'ESTM_NACIONALIDAD_COL_H' => $csvData[$i][23],
    'ESTM_NACIONALIDAD_COL_M' => $csvData[$i][24],
    'ESTM_NACIONALIDAD_ESP_H' => $csvData[$i][25],
    'ESTM_NACIONALIDAD_ESP_M' => $csvData[$i][26],
    'ESTM_NACIONALIDAD_FRA_H' => $csvData[$i][27],
    'ESTM_NACIONALIDAD_FRA_M' => $csvData[$i][28],
    'ESTM_NACIONALIDAD_USA_H' => $csvData[$i][29],
    'ESTM_NACIONALIDAD_USA_M' => $csvData[$i][30],
    'ESTM_NACIONALIDAD_PER_H' => $csvData[$i][31],
    'ESTM_NACIONALIDAD_PER_M' => $csvData[$i][32],
    'ESTM_NACIONALIDAD_RUM_H' => $csvData[$i][33],
    'ESTM_NACIONALIDAD_RUM_M' => $csvData[$i][34],
    'ESTM_NACIONALIDAD_CUB_H' => $csvData[$i][35],
    'ESTM_NACIONALIDAD_CUB_M' => $csvData[$i][36],
    'ESTM_NACIONALIDAD_URC_H' => $csvData[$i][37],
    'ESTM_NACIONALIDAD_URC_M' => $csvData[$i][38],
    'ESTM_NACIONALIDAD_VEN_H' => $csvData[$i][39],
    'ESTM_NACIONALIDAD_VEN_M' => $csvData[$i][40],
    'ESTM_DISCAPACIDAD_H' => $csvData[$i][40],
    'ESTM_DISCAPACIDAD_M' => $csvData[$i][41],
    'ESTM_TOTAL' => $csvData[$i][42],
    'ESTM_SEDE' => $csvData[$i][43],
    'ESTM_ESTADO' => $ESTM_ESTADO
];
```

- Recorremos el array y llenamos por cada columna acorde al valor correspondiente en cada columna

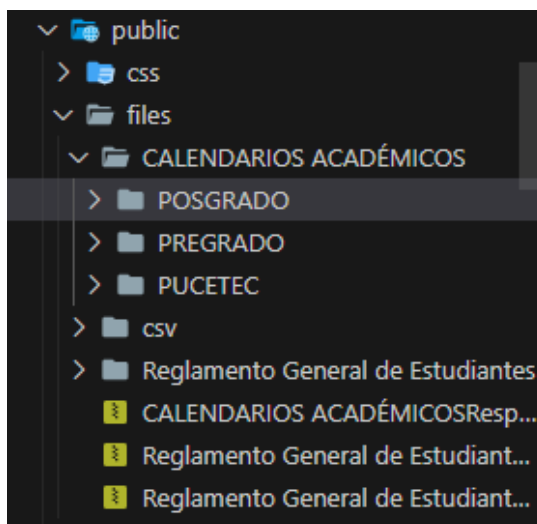
```
//subir a la bd
$objEstadMatr->insertar($datosprocesados[$i]);
}

//redireccionar
return redirect()->to(base_url() .
'index.php/subidaDatos/subirConjuntoDatos');
} else {
    echo "No se pudo abrir el archivo CSV.";
}
```

- Subimos el array mediante un modelo y redireccionamos

12. Codificación Ficheros

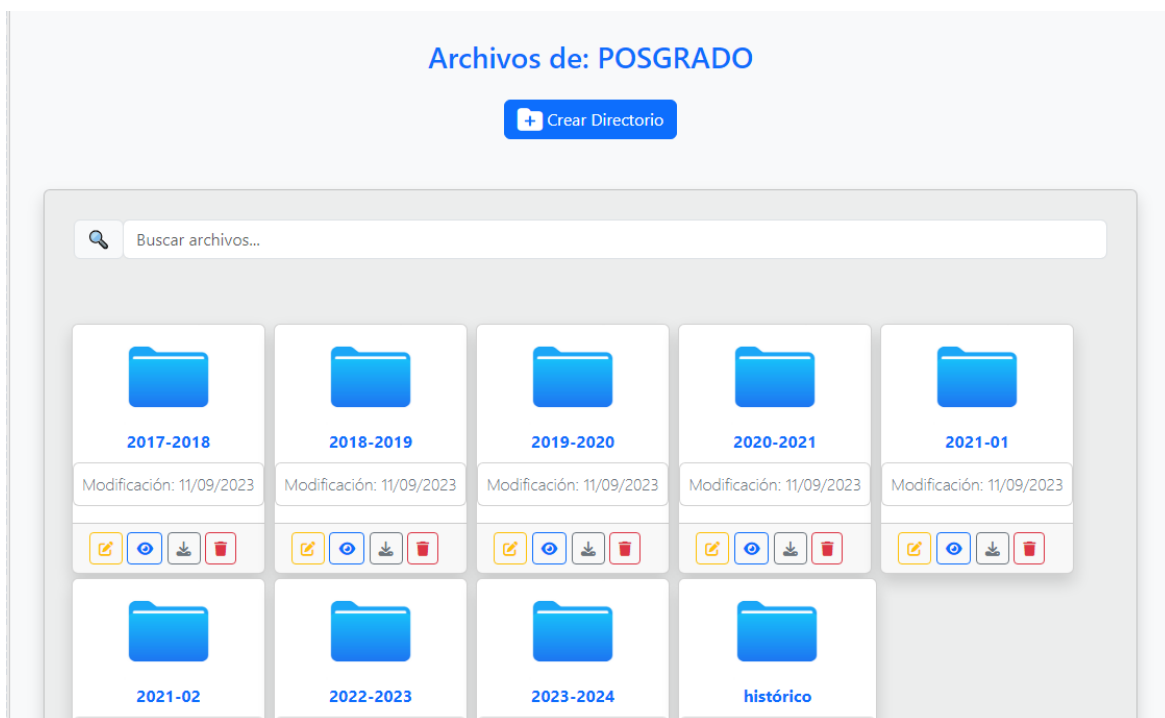
- Los ficheros corresponden a Calendario Académico y Reglamento general de Estudiantes, tomaremos de ejemplo a calendarios académicos
- El directorio esta almacenado en la ruta public/files, además cuenta con un rar en caso de recuperar algún archivo.



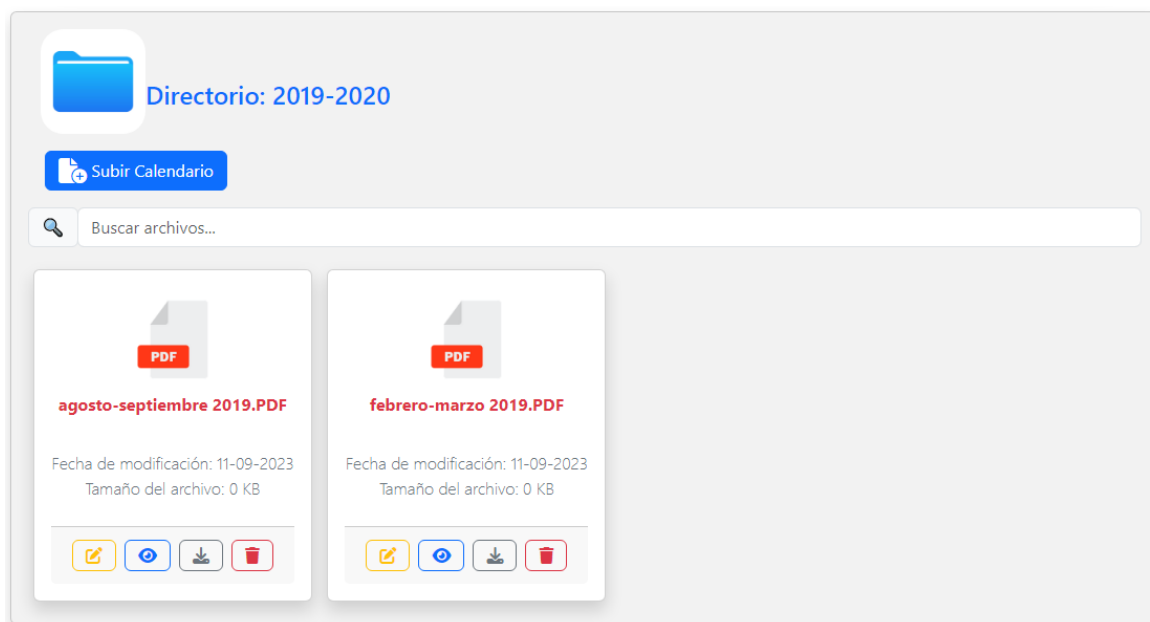
- Se recorre el fichero CALENDARIOS ACADÉMICOS y se genera la vista



3. Al acceder a cada carpeta se visualiza los archivos que contiene, además se puede crear ficheros, buscar dinámicamente y cada fichero cuenta con acciones de CRUD



- Dentro de cada fichero contiene archivos y se puede también subir archivos para organizar la información como corresponda, además de contar con un buscador y acciones CRUD dentro de la misma



4. ControladorCalendario académico

```
public function calendarioAcademico()
{
    try {

        // Directorio donde se encuentran los calendarios académicos
        $directorio =
'C:\XAMPP\htdocs\SistemaGestionDocumental\public\files\CALENDARIOS ACADÉMICOS';

        // Obtener la lista de archivos en el directorio
        $archivos = scandir($directorio);

        // Filtrar los archivos y directorios "." y ".."
        $archivos = array_diff($archivos, array('.', '..'));

        echo view('header');
        echo view('calendarioAcademico/calendarioAcademico', ['archivos' =>
$archivos]);
        echo view('footer');
    } catch (\Exception $e) {
        die($e->getMessage());
    }
}
```

- Se proporciona el directorio y se realizan las acciones dentro de cada controlador, en este caso se envía los ficheros para ver las carpetas que contiene el directorio

5. Para las acciones CRUD se sigue la lógica siguiente:

```

public function ver($nombre)
{
    try {
        $directorio = '';

        if ($nombre == 'POSGRADO') {
            $directorio =
'C:\XAMPP\htdocs\SistemaGestionDocumental\public\files\CALENDARIOS
ACADÉMICOS\POSGRADO'
;
        } elseif ($nombre == 'PREGRADO') {
            $directorio =
'C:\XAMPP\htdocs\SistemaGestionDocumental\public\files\CALENDARIOS
ACADÉMICOS\PREGRADO'
;
        } elseif ($nombre == 'PUCETEC') {
            $directorio =
'C:\XAMPP\htdocs\SistemaGestionDocumental\public\files\CALENDARIOS
ACADÉMICOS\PUCETEC'
;
        }

        // Obtener la lista de archivos y directorios en el directorio
        $archivos = scandir($directorio);

        // Filtrar los archivos y directorios "." y ".."
        $archivos = array_diff($archivos, array('.', '..'));

        echo view('header');
        echo view(
            'calendarioAcademico/ver',
            [
                'archivos' => $archivos,
                'nombre' => $nombre,
                'directorio' => $directorio
            ]
        );
        echo view('footer');
    } catch (\Exception $e) {
        die($e->getMessage());
    }
}

```

- Dependiendo de la búsqueda se va modificando el directorio, y en caso de CRUD sigue la misma lógica