

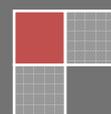
2013

# Encuesta de Actividades de innovación 2013

## Manual de Usuario

La Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) en su calidad de entidad rectora del Sistema Nacional de CTI, y consciente de su compromiso de impulsar la Ciencia, Tecnología, Innovación y los saberes ancestrales en el Ecuador, considera estratégico coordinar acciones de trabajo conjunto con el Instituto Nacional de Estadística y Censos (INEC); como el ente orientador y coordinador del Sistema Estadístico Nacional. Esta cooperación tiene como objetivo el levantamiento de información de Actividades de Ciencia, Tecnología e Innovación (ACTI) y el establecimiento de indicadores, que permitan obtener información validada y actualizada.

El 24 de julio del 2012 SENESCYT e INEC firman un convenio específico de cooperación técnica interinstitucional para el desarrollo del proyecto de la encuesta ACTI



## Índice

Índice .....	2
Introducción.....	6
Objetivo .....	6
Generalidades .....	6
Requerimientos del Sistema .....	7
Características mínimas del servidor:.....	7
Características mínimas del computador personal: .....	7
Requerimientos de software .....	7
Programas.....	7
ÍNDICE DE ELEMENTOS .....	8
Documentación de las estructuras de datos .....	14
Documentación de las funciones miembro .....	14
Referencia del Archivo: proyactinac/cal2.js.....	16
FUNCIONES.....	16
Documentación de las funciones .....	17
VARIABLES.....	20
Documentación de las variables .....	21
Referencia del Archivo: proyactinac/cal3.js.....	25
FUNCIONES.....	25
Documentación de las funciones .....	26
VARIABLES.....	29
Documentación de las variables .....	30
Referencia del Archivo: proyactinac/cobertura.php .....	33
VARIABLES.....	33
Documentación de las variables .....	33
Referencia del Archivo: proyactinac/conexion.php .....	34
VARIABLES.....	34
Documentación de las variables .....	34
Funciones.....	35

Documentación de las funciones .....	35
Variables .....	35
Documentación de las variables .....	35
Referencia del Archivo: proyactinac/excelwriter.inc.php .....	35
Funciones.....	35
Documentación de las funciones .....	36
Variables .....	36
Documentación de las variables .....	36
Referencia del Archivo: proyactinac/exportCambiosEncuesta.php .....	37
Variables .....	37
Documentación de las variables .....	37
Referencia del Archivo: proyactinac/exportCambiosMuestra.php .....	38
Variables .....	38
Documentación de las variables .....	39
Referencia del Archivo: proyactinac/exportDigitadores.php.....	40
Variables .....	40
Documentación de las variables .....	40
Referencia del Archivo: proyactinac/exportDigitadoresInc.php .....	41
Variables .....	41
Documentación de las variables .....	41
Referencia del Archivo: proyactinac/exportEncuesta.php.....	42
Documentación de las variables .....	43
Referencia del Archivo: proyactinac/funciones.js .....	43
FUNCIONES.....	43
Documentación de las funciones .....	45
VARIABLES.....	49
Documentación de las variables .....	50
Referencia del Archivo: proyactinac/index.php.....	53
FUNCIONES.....	53
Documentación de las funciones .....	53

VARIABLES.....	53
Documentación de las variables .....	53
Referencia del Archivo: proyactinac/index_respaldo.js .....	54
FUNCIONES.....	54
Documentación de las funciones .....	54
VARIABLES.....	54
Documentación de las variables .....	55
Referencia del Archivo: proyactinac/indexEnc.php .....	61
FUNCIONES.....	61
Documentación de las funciones .....	62
VARIABLES.....	62
Documentación de las variables .....	63
Referencia del Archivo: proyactinac/indexEnc2.php.....	68
FUNCIONES.....	68
Documentación de las funciones .....	68
VARIABLES.....	69
Documentación de las variables .....	70
Referencia del Archivo: proyactinac/indexRep.php .....	76
FUNCIONES.....	76
Documentación de las funciones .....	76
VARIABLES.....	76
Documentación de las variables .....	77
Referencia del Archivo: proyactinac/MMHTTPDB.php .....	78
FUNCIONES.....	78
Documentación de las funciones .....	79
VARIABLES.....	79
Documentación de las variables .....	79
ESTRUCTURAS DE DATOS.....	80
FUNCIONES.....	80
Documentación de las funciones .....	80

VARIABLES.....	80
----------------	----

## Introducción

Una vez desarrollado el sistema ACTI por la Unidad de Desarrollo y Administración de Sistemas Informáticos (DASI), se ha desarrollado el respectivo manual técnico, el cual indica cada uno de los procedimientos a seguir para el adecuado funcionamiento y manejo adecuado de dicha aplicación, aquí se pone a disposición del administrador o los posibles programadores la documentación técnica, la cual tiene como finalidad ofrecer el apoyo necesario para posibles modificaciones o mejoras en la misma, la documentación incluye los métodos utilizados, sus funciones y los archivos que han sido creados.

## Objetivo

Definir una línea base de investigación que permita obtener información sobre actividades de ciencia, tecnología e innovación (ACTI) y elaborar indicadores que permitan conocer los distintos aspectos de este proceso.

## Generalidades

HERRAMIENTAS DE APOYO	
LENGUAJE DE PROGRAMACIÓN	PHP
GESTOR DE BASE DE DATOS	Postgres
SERVIDOR WEB	Apache2
MANEJO DE ARCHIVOS DE APLICACIÓN	.xhtml .php .css .sql .xls .pdf
PLUGINS	Java

Tabla 1: Generalidades de la herramienta

## Requerimientos del Sistema

A continuación se detalla las características que debe poseer el servidor que contendrá la aplicación a la cual accederán las respectivas terminales.

### ***Características mínimas del servidor:***

Procesador	Intel® Core 2Duo 2.2 GHz
Memoria	2 Gb expandibles
Almacenamiento	Disco SAS de 300 GB
Comunicaciones	1 puerto Ethernet 100 Mbps
Periféricos	Lectora de DVD/CD, puerto USB

Tabla 2: Componentes del Servidor

Las características del computador del que se puede acceder al sistema para ingreso de información son las siguientes:

### ***Características mínimas del computador personal:***

Procesador	Intel® Core 2Duo 2.01 GHz
Memoria	512 MB
Almacenamiento	Disco Duro de 160 GB
Comunicaciones	1 puerto Ethernet 100 Mbps
Periféricos	Lectora de DVD/CD, puerto USB, teclado y mouse.

Tabla 3: Componentes del Pc personal

## Requerimientos de software

Para el correcto funcionamiento del sistema se debe poseer los siguientes programas instalados:

### ***Programas***

Sistema Operativo (32 ó 64 bits)	Windows, Linux o Mac
Plugins	Java
Navegador Web	Mozilla Firefox (recomendado), Internet Explorer, Opera.

Tabla 4: Software requerido

## ÍNDICE DE ELEMENTOS<sup>i</sup>

### a

**a1()** : [cal2.js](#) , [cal3.js](#)  
**addCalendar()** : [cal3.js](#) , [cal2.js](#)  
**ax()** : [cal3.js](#) , [cal2.js](#)

### b

**bandInn** : [funciones.js](#)  
**bandPreg2Sec3** : [funciones.js](#)  
**bandPregSec4** : [funciones.js](#)  
**bandS7S8** : [funciones.js](#)  
**bodyColor** : [cal2.js](#) , [cal3.js](#)  
**borderColor** : [cal3.js](#) , [cal2.js](#)

### c

**cal** : [cal2.js](#) , [cal3.js](#)  
**calBody()** : [cal3.js](#) , [cal2.js](#)  
**calFooter()** : [cal2.js](#) , [cal3.js](#)  
**calFormat** : [cal3.js](#) , [cal2.js](#)  
**calHeader()** : [cal2.js](#) , [cal3.js](#)  
**calHeight** : [cal2.js](#) , [cal3.js](#)  
**calOBJ()** : [cal2.js](#) , [cal3.js](#)  
**calOffsetX** : [cal3.js](#) , [cal2.js](#)  
**calOffsetY** : [cal2.js](#) , [cal3.js](#)  
**cals** : [cal2.js](#) , [cal3.js](#)  
**calTitle()** : [cal2.js](#) , [cal3.js](#)  
**calWidth** : [cal2.js](#) , [cal3.js](#)  
**calWin** : [cal2.js](#) , [cal3.js](#)  
**cargarPreg1\_4Sec10()** : [funciones.js](#)  
**cargarProvincias()** : [funciones.js](#)  
**changeCal()** : [cal2.js](#) , [cal3.js](#)  
**checkDate()** : [cal3.js](#) , [cal2.js](#)  
**clearDate()** : [cal2.js](#) , [cal3.js](#)  
**clickS10()** : [funciones.js](#)  
**clickS13()** : [funciones.js](#)  
**clickS6()** : [funciones.js](#)  
**clickS7()** : [funciones.js](#)  
**clickS8()** : [funciones.js](#)  
**close()** : [excelwriter.inc.php](#)  
**codRegion** : [funciones.js](#)  
**compareDates()** : [cal3.js](#) , [cal2.js](#)  
**conMayusculas()** : [funciones.js](#)  
**CONN\_NOT\_OPEN\_GET\_DB\_LIST** : [mysql.php](#)  
**CONN\_NOT\_OPEN\_GET\_TABLES** : [mysql.php](#)

---

**consultarCobertura() : funciones.js**  
**consultarDigitador() : funciones.js**  
**consultarEmpresa() : funciones.js**  
**consultarEncuesta() : funciones.js**  
**controlarSeccion15() : funciones.js**  
**crearComboPreg4Sec8() : funciones.js**  
**create\_error() : MMHTTPDB.php**  
**currentCal : cal2.js , cal3.js**  
**currentDayColor : cal2.js , cal3.js**  
**currentDayFontColor : cal3.js , cal2.js**

#### d

**dayColor : cal2.js , cal3.js**  
**dayDigits : cal3.js , cal2.js**  
**dayDisplay() : cal2.js , cal3.js**  
**dayFontColor : cal3.js , cal2.js**  
**daysColor : cal2.js , cal3.js**  
**daysFontColor : cal2.js , cal3.js**  
**dayWidth : cal3.js , cal2.js**

#### e

**else : index\_respaldo.php , indexEnc.php ,  
MMHTTPDB.php , indexEnc2.php**  
**errors1 : funciones.js**  
**errors10 : funciones.js**  
**errors11 : funciones.js**  
**errors12 : funciones.js**  
**errors13 : funciones.js**  
**errors14 : funciones.js**  
**errors15 : funciones.js**  
**errors1a : funciones.js**  
**errors2 : funciones.js**  
**errors3 : funciones.js**  
**errors4 : funciones.js**  
**errors5 : funciones.js**  
**errors6 : funciones.js**  
**errors7 : funciones.js**  
**errors8 : funciones.js**  
**errors9 : funciones.js**  
**errorsAd1 : funciones.js**  
**errorsAd2 : funciones.js**  
**ExcelWriter : excelwriter.inc.php**

#### f

**findCalendar() : cal2.js , cal3.js**

---

---

**firstDay()** : [cal3.js](#) , [cal2.js](#)  
**fontFace** : [cal2.js](#) , [cal3.js](#)  
**fontSize** : [cal3.js](#) , [cal2.js](#)  
**footColor** : [cal3.js](#) , [cal2.js](#)  
**footFontColor** : [cal2.js](#) , [cal3.js](#)

## g

**get2Digits()** : [cal2.js](#) , [cal3.js](#)  
**getCurrentDate()** : [cal3.js](#) , [cal2.js](#)  
**getDateNumbers()** : [cal2.js](#) , [cal3.js](#)  
**getDayName()** : [cal3.js](#) , [cal2.js](#)  
**GetFooter()** : [excelwriter.inc.php](#)  
**getFormat()** : [cal2.js](#) , [cal3.js](#)  
**GetHeader()** : [excelwriter.inc.php](#)  
**getLeftIE()** : [cal2.js](#) , [cal3.js](#)  
**getLeftN4()** : [cal3.js](#) , [cal2.js](#)  
**getLeftN6()** : [cal2.js](#) , [cal3.js](#)  
**getMonthFromName()** : [cal3.js](#) , [cal2.js](#)  
**getNumbers()** : [cal2.js](#) , [cal3.js](#)  
**getTopIE()** : [cal2.js](#) , [cal3.js](#)  
**getTopN4()** : [cal3.js](#) , [cal2.js](#)  
**getTopN6()** : [cal2.js](#) , [cal3.js](#)  
**guardarSeccion()** : [funciones.js](#)

## h

**hideCal()** : [cal2.js](#) , [cal3.js](#)

## i

**if** : [index.php](#) , [index\\_respaldo.php](#) , [indexEnc.php](#) ,  
[cal2.js](#) , [cal3.js](#) , [indexEnc.php](#) , [cal3.js](#) ,  
[index\\_respaldo.php](#) , [indexEnc2.php](#) , [index\\_respaldo.php](#)  
 , [indexEnc.php](#) , [indexEnc2.php](#) , [indexEnc.php](#) ,  
[indexRep.php](#) , [cal2.js](#)  
**is\_chrome()** : [index.php](#)  
**isAuthorized()** : [Error.php](#) , [index\\_respaldo.php](#) ,  
[indexEnc.php](#) , [indexRep.php](#) , [indexEnc2.php](#)  
**isIE** : [cal2.js](#) , [cal3.js](#)  
**isMac** : [cal3.js](#) , [cal2.js](#)  
**isN4** : [cal2.js](#) , [cal3.js](#)  
**isN6** : [cal3.js](#) , [cal2.js](#)  
**isOpera** : [cal2.js](#) , [cal3.js](#)  
**isOpera5** : [cal2.js](#) , [cal3.js](#)  
**isOpera6** : [cal2.js](#) , [cal3.js](#)

---

**k****KT\_ErrorHandler() : [mysql.php](#)****l****lastDay() : [cal2.js](#) , [cal3.js](#)****limpiarPreg1Sec6() : [funciones.js](#)****limpiarSec7() : [funciones.js](#)****limpiarSec8() : [funciones.js](#)****log\_messages() : [MMHTTPDB.php](#)****m****markClick() : [cal2.js](#) , [cal3.js](#)****MM\_validateForm() : [funciones.js](#)****modificarEncuesta() : [funciones.js](#)****modificarMuestra() : [funciones.js](#)****moveYear() : [cal2.js](#) , [cal3.js](#)****MYSQL\_NOT\_EXISTS : [mysql.php](#)****n****nav : [cal2.js](#) , [cal3.js](#)****nextMonth() : [cal3.js](#) , [cal2.js](#)****numeros() : [funciones.js](#)****o****onclick : [cal2.js](#) , [cal3.js](#)****open() : [excelwriter.inc.php](#)****p****pickDate() : [cal2.js](#) , [cal3.js](#)****prepMonth() : [cal3.js](#) , [cal2.js](#)****prueba() : [funciones.js](#)****q****quitarPreg2Seccion3\_1() : [funciones.js](#)****quitarPregSeccion11() : [funciones.js](#)****quitarPregSeccion12() : [funciones.js](#)****quitarPregSeccion15() : [funciones.js](#)****quitarPregSeccion4\_1() : [funciones.js](#)****quitarPregSeccion8() : [funciones.js](#)****quitarSeccion13() : [funciones.js](#)****r****radioButton : [funciones.js](#)****reporteActualizacion() : [funciones.js](#)****reporteDigitadores() : [funciones.js](#)**

---

return : [MMHTTPDB.php](#)

#### s

[setColor\(\)](#) : [cal2.js](#) , [cal3.js](#)  
[setDayNames\(\)](#) : [cal3.js](#) , [cal2.js](#)  
[setFont\(\)](#) : [cal2.js](#) , [cal3.js](#)  
[setFontColor\(\)](#) : [cal3.js](#) , [cal2.js](#)  
[setFormat\(\)](#) : [cal2.js](#) , [cal3.js](#)  
[setLinkNames\(\)](#) : [cal2.js](#) , [cal3.js](#)  
[setMonthNames\(\)](#) : [cal2.js](#) , [cal3.js](#)  
[setSize\(\)](#) : [cal3.js](#) , [cal2.js](#)  
[setWeekDay\(\)](#) : [cal2.js](#) , [cal3.js](#)  
[setWidth\(\)](#) : [cal2.js](#) , [cal3.js](#)  
[showCal\(\)](#) : [cal2.js](#) , [cal3.js](#)  
[soloNumero\(\)](#) : [funciones.js](#)  
[span1\(\)](#) : [cal2.js](#) , [cal3.js](#)  
[span2](#) : [cal3.js](#) , [cal2.js](#)  
[spanx\(\)](#) : [cal2.js](#) , [cal3.js](#)  
[sumaPreg1Sec14\(\)](#) : [funciones.js](#)  
[sumaPreg1Sec6\(\)](#) : [funciones.js](#)  
[sumaPreg3Sec3\(\)](#) : [funciones.js](#)  
[switchicon\(\)](#) : [switchicon.js](#)

#### t

[titleColor](#) : [cal2.js](#) , [cal3.js](#)  
[titleFontColor](#) : [cal3.js](#) , [cal2.js](#)  
[titleMode](#) : [cal2.js](#) , [cal3.js](#)  
[titleWidth](#) : [cal3.js](#) , [cal2.js](#)

#### v

[validarActAdminEnc\(\)](#) : [funciones.js](#)  
[validarActAdminEncEstadoC\(\)](#) : [funciones.js](#)  
[validarActAdminEncEstadoI\(\)](#) : [funciones.js](#)  
[validarActAdminMuestra\(\)](#) : [funciones.js](#)  
[validarActAdminMuestraEstadoA\(\)](#) : [funciones.js](#)  
[validarActAdminMuestraEstadoI\(\)](#) : [funciones.js](#)  
[validarPreg2Seccion3\(\)](#) : [funciones.js](#)  
[validarPreg3Seccion4\(\)](#) : [funciones.js](#)  
[validarRuc\(\)](#) : [funciones.js](#)  
[validarSeccion1\(\)](#) : [funciones.js](#)  
[validarSeccion10\(\)](#) : [funciones.js](#)  
[validarSeccion11\(\)](#) : [funciones.js](#)  
[validarSeccion12\(\)](#) : [funciones.js](#)  
[validarSeccion13\(\)](#) : [funciones.js](#)  
[validarSeccion14\(\)](#) : [funciones.js](#)

---

---

**validarSeccion15() : funciones.js**  
**validarSeccion2() : funciones.js**  
**validarSeccion3() : funciones.js**  
**validarSeccion4() : funciones.js**  
**validarSeccion5() : funciones.js**  
**validarSeccion6() : funciones.js**  
**validarSeccion7() : funciones.js**  
**validarSeccion8() : funciones.js**  
**validarSeccion9() : funciones.js**  
**validarTelefono() : funciones.js**  
**validarUso() : funciones.js**  
**verificarSec\_3\_4\_5() : funciones.js**

#### w

**weekDay : cal2.js , cal3.js**  
**winX : cal3.js , cal2.js**  
**winY : cal2.js , cal3.js**  
**writeCol() : excelwriter.inc.php**  
**writeLine() : excelwriter.inc.php**  
**writeRow() : excelwriter.inc.php**

#### y

**yxDays : cal2.js , cal3.js**  
**yxLinks : cal3.js , cal2.js**  
**yxMonths : cal3.js , cal2.js**

---

## Documentación de las estructuras de datos

### Referencia de la Clase Connection

#### MÉTODOS

- [Close \(\)](#)
- [ensureTicks \(\\$inputSQL\)](#)
- [ExecuteSP \(\\$aProcStatement, \\$TimeOut, \\$Parameters\)](#)
- [ExecuteSQL \(\\$aStatement, \\$MaxRows\)](#)
- [GetColumnsOfTable \(\\$TableName\)](#)
- [GetDatabaseList \(\)](#)
- [GetParametersOfProcedure \(\\$ProcedureName, \\$SchemaName, \\$CatalogName\)](#)
- [GetPrimaryKeysOfTable \(\\$TableName\)](#)
- [GetProcedures \(\)](#)
- [GetProviderTypes \(\)](#)
- [GetTables \(\\$table\\_name= ''\)](#)
- [GetViews \(\)](#)
- [HandleException \(\)](#)
- [MySqlConnection \(\\$ConnectionString, \\$Timeout, \\$Host, \\$DB, \\$UID, \\$Pwd\)](#)
- [Open \(\)](#)
- [ReturnsResultSet \(\\$ProcedureName\)](#)
- [SupportsProcedure \(\)](#)
- [TestOpen \(\)](#)

## Documentación de las funciones miembro

### Close ()

Definición en la línea 117 del archivo mysql.php.

### ensureTicks ( *\$inputSQL* )

Definición en la línea 336 del archivo mysql.php.  
Referenciado por GetTables().

**ExecuteSP ( *\$aProcStatement*, *\$TimeOut*, *\$Parameters* )**

Definición en la línea 310 del archivo mysql.php.

**ExecuteSQL ( *\$aStatement*, *\$MaxRows* )**

Definición en la línea 242 del archivo mysql.php.

**GetColumnsOfTable ( *\$TableName* )**

Definición en la línea 177 del archivo mysql.php.

**GetDatabaseList ( )**

Definición en la línea 356 del archivo mysql.php.

**GetParametersOfProcedure ( *\$ProcedureName*, *\$SchemaName*, *\$CatalogName* )**

Definición en la línea 236 del archivo mysql.php.

**GetPrimaryKeysOfTable ( *\$TableName* )**

Definición en la línea 376 del archivo mysql.php.

**GetProcedures ( )**

Definición en la línea 171 del archivo mysql.php.

**GetProviderTypes ( )**

Definición en la línea 305 del archivo mysql.php.

**GetTables ( *\$table\_name* = '' )**

Definición en la línea 129 del archivo mysql.php.

**GetViews ( )**

Definición en la línea 165 del archivo mysql.php.

**HandleException ( )**

Definición en la línea 328 del archivo mysql.php.

Referenciado por ExecuteSQL(), GetColumnsOfTable(), GetPrimaryKeysOfTable(), GetTables() y TestOpen().

**[MySQLConnection](#) ( *\$ConnectionString*, *\$Timeout*, *\$Host*, *\$DB*, *\$UID*, *\$Pwd* )**

Definición en la línea 64 del archivo mysql.php.

**Open ( )**

Definición en la línea 99 del archivo mysql.php.

**ReturnsResultSet ( *\$ProcedureName* )**

Definición en la línea 315 del archivo mysql.php.

**SupportsProcedure ( )**

Definición en la línea 320 del archivo mysql.php.

**TestOpen ( )**

Definición en la línea 112 del archivo mysql.php.

## Documentación de Archivos

# Referencia del Archivo: proyactinac/cal2.js

### **FUNCIONES**

- function [a1](#) (tag)
- function [addCalendar](#) (name, title, field, form)
- function [ax](#) (tag, color)
- function [calBody](#) (d, day)
- function [calFooter](#) ()
- function [calHeader](#) ()
- function [calOBJ](#) (name, title, field, form)
- function [calTitle](#) (d)
- function [changeCal](#) (d)
- function [checkDate](#) (name)
- function [clearDate](#) ()
- function [compareDates](#) (date1, date2)
- function [dayDisplay](#) (i)
- function [findCalendar](#) (name)
- function [firstDay](#) (d)
- function [get2Digits](#) (n)
- function [getCurrentDate](#) ()
- function [getDateNumbers](#) (date)
- function [getDayName](#) (y, m, d)
- function [getFormat](#) ()
- function [getLeftIE](#) (x, m)
- function [getLeftN4](#) (l)
- function [getLeftN6](#) (l)
- function [getMonthFromName](#) (m3)
- function [getNumbers](#) (date)
- function [getTopIE](#) (x, m)
- function [getTopN4](#) (l)
- function [getTopN6](#) (l)
- function [hideCal](#) ()
- [if](#) (isN4)
- [if](#) (isN4||isN6)
- function [lastDay](#) (d)
- function [markClick](#) (e)
- function [moveYear](#) (dy)
- function [nextMonth](#) (m)
- function [pickDate](#) (d)
- function [prepMonth](#) (m)
- function [setColor](#) (tColor, dsColor, bColor, dColor, cdColor, fColor, bdColor)
- function [setDayNames](#) (sunName, monName, tueName, wedName, thuName, friName, satName)
- function [setFont](#) (font, size)
- function [setFontColor](#) (tColorFont, dsColorFont, dColorFont, cdColorFont, fColorFont)
- function [setFormat](#) (format)
- function [setLinkNames](#) (closeLink, clearLink)
- function [setMonthNames](#) (janName, febName, marName, aprName, mayName, junName, julName, augName, sepName, octName, novName, decName)
- function [setSize](#) (width, height, ox, oy)
- function [setWeekDay](#) (wDay)
- function [setWidth](#) (tWidth, tMode, dWidth, dDigits)
- function [showCal](#) (name)
- function [span1](#) (tag)

- function [spanx](#) (tag, color)

### **Documentación de las funciones**

#### **function a1 ( tag )**

Definición en la línea 75 del archivo cal2.js.  
Referenciado por calBody(), calFooter() y calTitle().

#### **function addCalendar ( name, title, field, form )**

Definición en la línea 256 del archivo cal2.js.

#### **function ax ( tag, color )**

Definición en la línea 79 del archivo cal2.js.  
Referenciado por calHeader().

#### **function calBody ( d, day )**

Definición en la línea 433 del archivo cal2.js.  
Referenciado por changeCal() y showCal().

#### **function calFooter ( )**

Definición en la línea 429 del archivo cal2.js.  
Referenciado por changeCal() y showCal().

#### **function calHeader ( )**

Definición en la línea 425 del archivo cal2.js.  
Referenciado por changeCal() y showCal().

#### **function calOBJ ( name, title, field, form )**

Definición en la línea 83 del archivo cal2.js.  
Referenciado por addCalendar().

#### **function calTitle ( d )**

Definición en la línea 405 del archivo cal2.js.  
Referenciado por changeCal() y showCal().

#### **function changeCal ( d )**

Definición en la línea 485 del archivo cal2.js.  
Referenciado por moveYear(), nextMonth() y prepMonth().

#### **function checkDate ( name )**

Definición en la línea 625 del archivo cal2.js.

#### **function clearDate ( )**

Definición en la línea 600 del archivo cal2.js.

#### **function compareDates ( date1, date2 )**

Definición en la línea 652 del archivo cal2.js.

#### **function dayDisplay ( i )**

Definición en la línea 396 del archivo cal2.js.  
Referenciado por calTitle().

**function findCalendar ( *name* )**

Definición en la línea 260 del archivo cal2.js.  
Referenciado por checkDate() y showCal().

**function firstDay ( *d* )**

Definición en la línea 390 del archivo cal2.js.  
Referenciado por calBody().

**function get2Digits ( *n* )**

Definición en la línea 596 del archivo cal2.js.  
Referenciado por getCurrentDate() y pickDate().

**function getCurrentDate ( )**

Definición en la línea 643 del archivo cal2.js.  
Referenciado por compareDates().

**function getDateNumbers ( *date* )**

Definición en la línea 325 del archivo cal2.js.  
Referenciado por changeCal() y showCal().

**function getDayName ( *y, m, d* )**

Definición en la línea 281 del archivo cal2.js.  
Referenciado por pickDate().

**function getFormat ( )**

Definición en la línea 296 del archivo cal2.js.  
Referenciado por changeCal(), checkDate(), compareDates(), getNumbers() y showCal().

**function getLeftIE ( *x, m* )**

Definición en la línea 351 del archivo cal2.js.  
Referenciado por getLeftIE().

**function getLeftN4 ( *l* )**

Definición en la línea 374 del archivo cal2.js.

**function getLeftN6 ( *l* )**

Definición en la línea 377 del archivo cal2.js.

**function getMonthFromName ( *m3* )**

Definición en la línea 286 del archivo cal2.js.  
Referenciado por getDateNumbers() y getNumbers().

**function getNumbers ( *date* )**

Definición en la línea 684 del archivo cal2.js.  
Referenciado por compareDates().

**function getTopIE ( *x, m* )**

Definición en la línea 362 del archivo cal2.js.  
Referenciado por getTopIE().

**function getTopN4 ( *l* )**

Definición en la línea 375 del archivo cal2.js.

**function getTopN6 ( *l* )**

Definición en la línea 378 del archivo cal2.js.

**function hideCal ( )**

Definición en la línea 345 del archivo cal2.js.

Referenciado por clearDate() y pickDate().

**if ( *isN4* )**

Definición en la línea 62 del archivo cal2.js.

**if ( *isN4* || *isN6* )**

Definición en la línea 712 del archivo cal2.js.

**function lastDay ( *d* )**

Definición en la línea 380 del archivo cal2.js.

Referenciado por calBody().

**function markClick ( *e* )**

Definición en la línea 512 del archivo cal2.js.

**function moveYear ( *dy* )**

Definición en la línea 465 del archivo cal2.js.

**function nextMonth ( *m* )**

Definición en la línea 478 del archivo cal2.js.

**function pickDate ( *d* )**

Definición en la línea 605 del archivo cal2.js.

**function prepMonth ( *m* )**

Definición en la línea 471 del archivo cal2.js.

**function setColor ( *tColor, dsColor, bColor, dColor, cdColor, fColor, bdColor* )**

Definición en la línea 119 del archivo cal2.js.

**function setDayNames ( *sunName, monName, tueName, wedName, thuName, friName, satName* )**

Definición en la línea 222 del archivo cal2.js.

**function setFont ( *font, size* )**

Definición en la línea 91 del archivo cal2.js.

**function setFontColor ( *tColorFont, dsColorFont, dColorFont, cdColorFont, fColorFont* )**

Definición en la línea 143 del archivo cal2.js.

**function setFormat ( *format* )**

Definición en la línea 161 del archivo cal2.js.

**function setLinkNames ( *closeLink, clearLink* )**

Definición en la línea 247 del archivo cal2.js.

**function setMonthNames ( *janName, febName, marName, aprName, mayName, junName, julName, augName, sepName, octName, novName, decName* )**

Definición en la línea 183 del archivo cal2.js.

**function setSize ( *width, height, ox, oy* )**

Definición en la línea 165 del archivo cal2.js.

**function setWeekDay ( *wDay* )**

Definición en la línea 177 del archivo cal2.js.

**function setWidth ( *tWidth, tMode, dWidth, dDigits* )**

Definición en la línea 104 del archivo cal2.js.

**function showCal ( *name* )**

Definición en la línea 534 del archivo cal2.js.

**function span1 ( *tag* )**

Definición en la línea 68 del archivo cal2.js.

Referenciado por calBody(), calFooter() y calTitle().

**function spanx ( *tag, color* )**

Definición en la línea 71 del archivo cal2.js.

Referenciado por calHeader().

## VARIABLES

- var [bodyColor](#) ="#ffffff"
- var [borderColor](#) ="#333333"
- var [cal](#) ="cal"
- var [calFormat](#) = "yyyy-mm-dd"
- var [calHeight](#) =200
- var [calOffsetX](#) =-200
- var [calOffsetY](#) =16
- var [cals](#) =new Array()
- var [calWidth](#) =200
- var [calWin](#) =null
- var [currentCal](#) =null
- var [currentDayColor](#) ="#333333"
- var [currentDayFontColor](#) = "#ffffff"
- var [dayColor](#) ="#ffffff"
- var [dayDigits](#) =1
- var [dayFontColor](#) = "#333333"
- var [daysColor](#) ="#cccccc"
- var [daysFontColor](#) = "#333333"
- var [dayWidth](#) =12
- var [fontFace](#) ="verdana"
- var [fontSize](#) =9

- var `footColor` = "#cccccc"
- var `footFontColor` = "#333333"
- var `isIE` =(document.all && !`isOpera` && (!`isMac` || navigator.appVersion.indexOf("MSIE 4")===-1))?true:false
- var `isMac` =(nav.indexOf("mac")!=-1)
- var `isN4` =(document.layers)?true:false
- var `isN6` =(nav.indexOf("gecko")!=-1)
- var `isOpera` =(nav.indexOf("opera")!=-1)?true:false
- var `isOpera5` =(nav.indexOf("opera 5")!=-1 || nav.indexOf("opera/5")!=-1)?true:false
- var `isOpera6` =(`isOpera` && parseInt(navigator.appVersion)>=6)?true:false
- var `nav` =navigator.userAgent.toLowerCase()
- document `onclick` =`markClick`
- var `span2` = "</span>"
- var `titleColor` = "#cccccc"
- var `titleFontColor` = "#333333"
- var `titleMode` =1
- var `titleWidth` =90
- var `weekDay` = 0
- var `winX` =0
- var `winY` =0
- var `yxDays` =new Array("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
- var `yxLinks` =new Array("[close]", "[clear]")
- var `yxMonths` =new Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")

### ***Documentación de las variables***

**var `bodyColor` = "#ffffff"**

Definición en la línea 24 del archivo cal2.js.  
Referenciado por calBody() y setColor().

**var `borderColor` = "#333333"**

Definición en la línea 28 del archivo cal2.js.  
Referenciado por calHeader() y setColor().

**var `cal` = "cal"**

Definición en la línea 45 del archivo cal2.js.

**var `calFormat` = "yyyy-mm-dd"**

Definición en la línea 36 del archivo cal2.js.  
Referenciado por getCurrentDate(), getDateNumbers(), getFormat(), getNumbers(), pickDate() y setFormat().

**var `calHeight` = 200**

Definición en la línea 42 del archivo cal2.js.  
Referenciado por setSize() y showCal().

**var calOffsetX = -200**

Definición en la línea 42 del archivo cal2.js.  
Referenciado por setSize() y showCal().

**var calOffsetY = 16**

Definición en la línea 42 del archivo cal2.js.  
Referenciado por setSize() y showCal().

**var cal = new Array()**

Definición en la línea 46 del archivo cal2.js.  
Referenciado por addCalendar() y findCalendar().

**var calWidth = 200**

Definición en la línea 42 del archivo cal2.js.  
Referenciado por setSize() y showCal().

**var calWin = null**

Definición en la línea 43 del archivo cal2.js.  
Referenciado por changeCal(), hideCal() y showCal().

**var currentCal = null**

Definición en la línea 47 del archivo cal2.js.  
Referenciado por calHeader(), changeCal(), clearDate(), pickDate() y showCal().

**var currentDayColor = "#333333"**

Definición en la línea 26 del archivo cal2.js.  
Referenciado por calBody() y setColor().

**var currentDayFontColor = "#ffffff"**

Definición en la línea 33 del archivo cal2.js.  
Referenciado por calHeader() y setFontColor().

**var dayColor = "#ffffff"**

Definición en la línea 25 del archivo cal2.js.  
Referenciado por calBody() y setColor().

**var dayDigits = 1**

Definición en la línea 20 del archivo cal2.js.  
Referenciado por dayDisplay() y setWidth().

**var dayFontColor = "#333333"**

Definición en la línea 32 del archivo cal2.js.  
Referenciado por calHeader() y setFontColor().

**var daysColor = "#cccccc"**

Definición en la línea 23 del archivo cal2.js.  
Referenciado por calTitle() y setColor().

**var daysFontColor = "#333333"**

Definición en la línea 31 del archivo cal2.js.  
Referenciado por calHeader() y setFontColor().

**var dayWidth =12**

Definición en la línea 19 del archivo cal2.js.  
Referenciado por calTitle() y setWidth().

**var fontFace ="verdana"**

Definición en la línea 14 del archivo cal2.js.  
Referenciado por setFont() y spanx().

**var fontSize =9**

Definición en la línea 15 del archivo cal2.js.  
Referenciado por if(), setFont() y spanx().

**var footColor ="#cccccc"**

Definición en la línea 27 del archivo cal2.js.  
Referenciado por calFooter(), calHeader() y setColor().

**var footFontColor = "#333333"**

Definición en la línea 34 del archivo cal2.js.  
Referenciado por calHeader() y setFontColor().

**var isIE =(document.all && !isOpera && (!isMac || navigator.appVersion.indexOf("MSIE 4")!=-1))?true:false**

Definición en la línea 60 del archivo cal2.js.

**var isMac =(nav.indexOf("mac")!=-1)**

Definición en la línea 59 del archivo cal2.js.

**var isN4 =(document.layers)?true:false**

Definición en la línea 58 del archivo cal2.js.

**var isN6 =(nav.indexOf("gecko")!=-1)**

Definición en la línea 57 del archivo cal2.js.

**var isOpera =(nav.indexOf("opera")!=-1)?true:false**

Definición en la línea 54 del archivo cal2.js.

**var isOpera5 =(nav.indexOf("opera 5")!=-1 || nav.indexOf("opera/5")!=-1)?true:false**

Definición en la línea 55 del archivo cal2.js.

**var isOpera6 =(isOpera && parseInt(navigator.appVersion)>=6)?true:false**

Definición en la línea 56 del archivo cal2.js.

**var nav =navigator.userAgent.toLowerCase()**

Definición en la línea 53 del archivo cal2.js.

**document onclick =markClick**

Definición en la línea 715 del archivo cal2.js.

**var span2 ="</span>"**

Definición en la línea 66 del archivo cal2.js.

**var titleColor = "#cccccc"**

Definición en la línea 22 del archivo cal2.js.

Referenciado por calTitle() y setColor().

**var titleFontColor = "#333333"**

Definición en la línea 30 del archivo cal2.js.

Referenciado por calHeader() y setFontColor().

**var titleMode = 1**

Definición en la línea 18 del archivo cal2.js.

Referenciado por calTitle() y setWidth().

**var titleWidth = 90**

Definición en la línea 17 del archivo cal2.js.

Referenciado por calTitle() y setWidth().

**var weekDay = 0**

Definición en la línea 38 del archivo cal2.js.

Referenciado por calBody(), calTitle() y setWeekDay().

**var winX = 0**

Definición en la línea 44 del archivo cal2.js.

Referenciado por markClick() y showCal().

**var winY = 0**

Definición en la línea 44 del archivo cal2.js.

Referenciado por markClick() y showCal().

**var yxDays = new Array("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",  
"Friday", "Saturday", "Sunday")**

Definición en la línea 50 del archivo cal2.js.

Referenciado por dayDisplay(), getDayName() y setDayNames().

**var yxLinks = new Array("[close]", "[clear]")**

Definición en la línea 51 del archivo cal2.js.

Referenciado por calFooter() y setLinkNames().

**var yxMonths = new Array("January", "February", "March", "April", "May", "June",  
"July", "August", "September", "October", "November", "December")**

Definición en la línea 49 del archivo cal2.js.

Referenciado por calTitle(), getMonthFromName(), pickDate() y setMonthNames().

## Referencia del Archivo: proyactinac/cal3.js

### FUNCIONES

- function [a1](#) (tag)
- function [addCalendar](#) (name, title, field, form)
- function [ax](#) (tag, color)
- function [calBody](#) (d, day)
- function [calFooter](#) ()
- function [calHeader](#) ()
- function [calOBJ](#) (name, title, field, form)
- function [calTitle](#) (d)
- function [changeCal](#) (d)
- function [checkDate](#) (name)
- function [clearDate](#) ()
- function [compareDates](#) (date1, date2)
- function [dayDisplay](#) (i)
- function [findCalendar](#) (name)
- function [firstDay](#) (d)
- function [get2Digits](#) (n)
- function [getCurrentDate](#) ()
- function [getDateNumbers](#) (date)
- function [getDayName](#) (y, m, d)
- function [getFormat](#) ()
- function [getLeftIE](#) (x, m)
- function [getLeftN4](#) (l)
- function [getLeftN6](#) (l)
- function [getMonthFromName](#) (m3)
- function [getNumbers](#) (date)
- function [getTopIE](#) (x, m)
- function [getTopN4](#) (l)
- function [getTopN6](#) (l)
- function [hideCal](#) ()
- [if \(isN4\)](#)
- [if \(isN4||isN6\)](#)
- function [lastDay](#) (d)
- function [markClick](#) (e)
- function [moveYear](#) (dy)
- function [nextMonth](#) (m)
- function [pickDate](#) (d)
- function [prepMonth](#) (m)
- function [setColor](#) (tColor, dsColor, bColor, dColor, cdColor, fColor, bdColor)
- function [setDayNames](#) (sunName, monName, tueName, wedName, thuName, friName, satName)
- function [setFont](#) (font, size)
- function [setFontColor](#) (tColorFont, dsColorFont, dColorFont, cdColorFont, fColorFont)
- function [setFormat](#) (format)
- function [setLinkNames](#) (closeLink, clearLink)
- function [setMonthNames](#) (janName, febName, marName, aprName, mayName, junName, julName, augName, sepName, octName, novName, decName)
- function [setSize](#) (width, height, ox, oy)
- function [setWeekDay](#) (wDay)
- function [setWidth](#) (tWidth, tMode, dWidth, dDigits)
- function [showCal](#) (name)
- function [span1](#) (tag)
- function [spanx](#) (tag, color)

### ***Documentación de las funciones***

#### **function a1 ( *tag* )**

Definición en la línea 75 del archivo cal3.js.

#### **function addCalendar ( *name, title, field, form* )**

Definición en la línea 256 del archivo cal3.js.

#### **function ax ( *tag, color* )**

Definición en la línea 79 del archivo cal3.js.

#### **function calBody ( *d, day* )**

Definición en la línea 433 del archivo cal3.js.

#### **function calFooter ( )**

Definición en la línea 429 del archivo cal3.js.

#### **function calHeader ( )**

Definición en la línea 425 del archivo cal3.js.

#### **function calOBJ ( *name, title, field, form* )**

Definición en la línea 83 del archivo cal3.js.

#### **function calTitle ( *d* )**

Definición en la línea 405 del archivo cal3.js.

#### **function changeCal ( *d* )**

Definición en la línea 485 del archivo cal3.js.

#### **function checkDate ( *name* )**

Definición en la línea 625 del archivo cal3.js.

#### **function clearDate ( )**

Definición en la línea 600 del archivo cal3.js.

#### **function compareDates ( *date1, date2* )**

Definición en la línea 652 del archivo cal3.js.

#### **function dayDisplay ( *i* )**

Definición en la línea 396 del archivo cal3.js.

#### **function findCalendar ( *name* )**

Definición en la línea 260 del archivo cal3.js.

#### **function firstDay ( *d* )**

Definición en la línea 390 del archivo cal3.js.

**function get2Digits ( *n* )**

Definición en la línea 596 del archivo cal3.js.

**function getCurrentDate ( )**

Definición en la línea 643 del archivo cal3.js.

**function getDateNumbers ( *date* )**

Definición en la línea 325 del archivo cal3.js.

**function getDayName ( *y, m, d* )**

Definición en la línea 281 del archivo cal3.js.

**function getFormat ( )**

Definición en la línea 296 del archivo cal3.js.

**function getLeftIE ( *x, m* )**

Definición en la línea 351 del archivo cal3.js.

**function getLeftN4 ( *l* )**

Definición en la línea 374 del archivo cal3.js.

**function getLeftN6 ( *l* )**

Definición en la línea 377 del archivo cal3.js.

**function getMonthFromName ( *m3* )**

Definición en la línea 286 del archivo cal3.js.

**function getNumbers ( *date* )**

Definición en la línea 684 del archivo cal3.js.

**function getTopIE ( *x, m* )**

Definición en la línea 362 del archivo cal3.js.

**function getTopN4 ( *l* )**

Definición en la línea 375 del archivo cal3.js.

**function getTopN6 ( *l* )**

Definición en la línea 378 del archivo cal3.js.

**function hideCal ( )**

Definición en la línea 345 del archivo cal3.js.

**if ( *isN4* )**

Definición en la línea 62 del archivo cal3.js.

**if ( [isN4](#) | | *isN6* )**

Definición en la línea 712 del archivo cal3.js.

**function lastDay ( *d* )**

Definición en la línea 380 del archivo cal3.js.

**function markClick ( e )**

Definición en la línea 512 del archivo cal3.js.

**function moveYear ( dy )**

Definición en la línea 465 del archivo cal3.js.

**function nextMonth ( m )**

Definición en la línea 478 del archivo cal3.js.

**function pickDate ( d )**

Definición en la línea 605 del archivo cal3.js.

**function prepMonth ( m )**

Definición en la línea 471 del archivo cal3.js.

**function setColor ( tColor, dsColor, bColor, dColor, cdColor, fColor, bdColor )**

Definición en la línea 119 del archivo cal3.js.

**function setDayNames ( sunName, monName, tueName, wedName, thuName, friName, satName )**

Definición en la línea 222 del archivo cal3.js.

**function setFont ( font, size )**

Definición en la línea 91 del archivo cal3.js.

**function setFontColor ( tColorFont, dsColorFont, dColorFont, cdColorFont, fColorFont )**

Definición en la línea 143 del archivo cal3.js.

**function setFormat ( format )**

Definición en la línea 161 del archivo cal3.js.

**function setLinkNames ( closeLink, clearLink )**

Definición en la línea 247 del archivo cal3.js.

**function setMonthNames ( janName, febName, marName, aprName, mayName, junName, julName, augName, sepName, octName, novName, decName )**

Definición en la línea 183 del archivo cal3.js.

**function setSize ( width, height, ox, oy )**

Definición en la línea 165 del archivo cal3.js.

**function setWeekDay ( *wDay* )**

Definición en la línea 177 del archivo cal3.js.

**function setWidth ( *tWidth, tMode, dWidth, dDigits* )**

Definición en la línea 104 del archivo cal3.js.

**function showCal ( *name* )**

Definición en la línea 534 del archivo cal3.js.

**function span1 ( *tag* )**

Definición en la línea 68 del archivo cal3.js.

**function spanx ( *tag, color* )**

Definición en la línea 71 del archivo cal3.js.

**VARIABLES**

- var [bodyColor](#) ="#ffffff"
- var [borderColor](#) ="#333333"
- var [cal](#) ="cal"
- var [calFormat](#) = "yyyy-mm-dd"
- var [calHeight](#) =200
- var [calOffsetX](#) =-200
- var [calOffsetY](#) =16
- var [cals](#) =new Array()
- var [calWidth](#) =200
- var [calWin](#) =null
- var [currentCal](#) =null
- var [currentDayColor](#) ="#333333"
- var [currentDayFontColor](#) = "#ffffff"
- var [dayColor](#) ="#ffffff"
- var [dayDigits](#) =1
- var [dayFontColor](#) = "#333333"
- var [daysColor](#) ="#cccccc"
- var [daysFontColor](#) = "#333333"
- var [dayWidth](#) =12
- var [fontFace](#) ="verdana"
- var [fontSize](#) =9
- var [footColor](#) ="#cccccc"
- var [footFontColor](#) = "#333333"
- var [isIE](#) =(document.all && ![isOpera](#) && (![isMac](#) || navigator.appVersion.indexOf("MSIE 4")!=-1))?true:false
- var [isMac](#) =(nav.indexOf("mac")!=-1)
- var [isN4](#) =(document.layers)?true:false
- var [isN6](#) =(nav.indexOf("gecko")!=-1)

- var `isOpera` =(nav.indexOf("opera")!=-1)?true:false
- var `isOpera5` =(nav.indexOf("opera 5")!=-1 || nav.indexOf("opera/5")!=-1)?true:false
- var `isOpera6` =(isOpera && parseInt(navigator.appVersion)>=6)?true:false
- var `nav` =navigator.userAgent.toLowerCase()
- document `onclick` =`markClick`
- var `span2` ="</span>"
- var `titleColor` ="#cccccc"
- var `titleFontColor` ="#333333"
- var `titleMode` =1
- var `titleWidth` =90
- var `weekDay` = 0
- var `winX` =0
- var `winY` =0
- var `yxDays` =new Array("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
- var `yxLinks` =new Array("[close]", "[clear]")
- var `yxMonths` =new Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")

### ***Documentación de las variables***

**var bodyColor = "#ffffff"**

Definición en la línea 24 del archivo cal3.js.

**var borderColor = "#333333"**

Definición en la línea 28 del archivo cal3.js.

**var cal = "cal"**

Definición en la línea 45 del archivo cal3.js.

**var calFormat = "yyyy-mm-dd"**

Definición en la línea 36 del archivo cal3.js.

**var calHeight = 200**

Definición en la línea 42 del archivo cal3.js.

**var calOffsetX = -200**

Definición en la línea 42 del archivo cal3.js.

**var calOffsetY = 16**

Definición en la línea 42 del archivo cal3.js.  
**var calS = new Array()**

Definición en la línea 46 del archivo cal3.js.  
**var calWidth = 200**

Definición en la línea 42 del archivo cal3.js.  
**var calWin = null**

Definición en la línea 43 del archivo cal3.js.  
**var currentCal = null**

Definición en la línea 47 del archivo cal3.js.  
**var currentDayColor = "#333333"**

Definición en la línea 26 del archivo cal3.js.  
**var currentDayFontColor = "#ffffff"**

Definición en la línea 33 del archivo cal3.js.  
**var dayColor = "#ffffff"**

Definición en la línea 25 del archivo cal3.js.  
**var dayDigits = 1**

Definición en la línea 20 del archivo cal3.js.  
**var dayFontColor = "#333333"**

Definición en la línea 32 del archivo cal3.js.  
**var daysColor = "#cccccc"**

Definición en la línea 23 del archivo cal3.js.  
**var daysFontColor = "#333333"**

Definición en la línea 31 del archivo cal3.js.  
**var dayWidth = 12**

Definición en la línea 19 del archivo cal3.js.  
**var fontFace = "verdana"**

Definición en la línea 14 del archivo cal3.js.  
**var fontSize = 9**

Definición en la línea 15 del archivo cal3.js.  
**var footColor = "#cccccc"**

Definición en la línea 27 del archivo cal3.js.  
**var footFontColor = "#333333"**

Definición en la línea 34 del archivo cal3.js.

```
var isIE = (document.all && !isOpera && (!isMac || navigator.appVersion.indexOf("MSIE 4")!=-1))?true:false
```

Definición en la línea 60 del archivo cal3.js.

```
var isMac = (nav.indexOf("mac")!=-1)
```

Definición en la línea 59 del archivo cal3.js.

```
var isN4 = (document.layers)?true:false
```

Definición en la línea 58 del archivo cal3.js.

```
var isN6 = (nav.indexOf("gecko")!=-1)
```

Definición en la línea 57 del archivo cal3.js.

```
var isOpera = (nav.indexOf("opera")!=-1)?true:false
```

Definición en la línea 54 del archivo cal3.js.

```
var isOpera5 = (nav.indexOf("opera 5")!=-1 || nav.indexOf("opera/5")!=-1)?true:false
```

Definición en la línea 55 del archivo cal3.js.

```
var isOpera6 = (isOpera && parseInt(navigator.appVersion)>=6)?true:false
```

Definición en la línea 56 del archivo cal3.js.

```
var nav = navigator.userAgent.toLowerCase()
```

Definición en la línea 53 del archivo cal3.js.

```
document onclick = markClick
```

Definición en la línea 715 del archivo cal3.js.

```
var span2 = "</span>"
```

Definición en la línea 66 del archivo cal3.js.

```
var titleColor = "#cccccc"
```

Definición en la línea 22 del archivo cal3.js.

```
var titleFontColor = "#333333"
```

Definición en la línea 30 del archivo cal3.js.

```
var titleMode = 1
```

Definición en la línea 18 del archivo cal3.js.

```
var titleWidth = 90
```

Definición en la línea 17 del archivo cal3.js.

```
var weekDay = 0
```

Definición en la línea 38 del archivo cal3.js.

```
var winX = 0
```

Definición en la línea 44 del archivo cal3.js.

**var winY =0**

Definición en la línea 44 del archivo cal3.js.

**var yxDays =new Array("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")**

Definición en la línea 50 del archivo cal3.js.

**var yxLinks =new Array("[close]", "[clear]")**

Definición en la línea 51 del archivo cal3.js.

**var yxMonths =new Array("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")**

Definición en la línea 49 del archivo cal3.js.

## Referencia del Archivo: proyactinac/cobertura.php

### VARIABLES

- [\\$encuesta](#) = pg\_query(\$conexion, \$query\_encuesta) or die('Error en la consulta SQL cobertura encuesta'. pg\_last\_error(\$conexion))
- [if](#)(isset(\$\_REQUEST["regional"])&&\$\_REQUEST["regional"]!="")
- [if](#)((isset(\$\_REQUEST["fecha\_registro"])&&\$\_REQUEST["fecha\_registro"]!="")&&(isset(\$\_REQUEST["fecha\_registro2"])&&\$\_REQUEST["fecha\_registro2"]!="")) [\\$query\\_encuesta](#) = sprintf("SELECT \* FROM encuesta ".\$where." AND estado = 'C'")
- [\\$row\\_encuesta](#) = pg\_fetch\_assoc(\$encuesta)
- [\\$totalRows\\_encuesta](#) = pg\_num\_rows(\$encuesta)
- [\\$where](#) = " WHERE seccion >= 16"

### Documentación de las variables

**\$encuesta = pg\_query(\$conexion, \$query\_encuesta) or die('Error en la consulta SQL cobertura encuesta'. pg\_last\_error(\$conexion))**

Definición en la línea 33 del archivo cobertura.php.

```
if      (isset($_REQUEST["regional"])&&$_REQUEST["regional"]!="")      if
((isset($_REQUEST["fecha_registro"])&&$_REQUEST["fecha_registro"]!="")&&(isset($_REQUEST["fecha_registro2"])&&$_REQUEST["fecha_registro2"]!=""))
$query_encuesta = sprintf("SELECT * FROM encuesta ".$where." AND estado = 'C')
```

Definición en la línea 32 del archivo cobertura.php.

**\$row\_encuesta = pg\_fetch\_assoc(\$encuesta)**

Definición en la línea 34 del archivo cobertura.php.

```
$totalRows_encuesta = pg_num_rows($encuesta)
```

Definición en la línea 35 del archivo cobertura.php.

```
$where = " WHERE seccion >= 16"
```

Definición en la línea 5 del archivo cobertura.php.

## Referencia del Archivo: proyactinac/conexion.php

### VARIABLES

- `$conexion` = `pg_connect("host=localhost dbname=acti_nacional user=postgres password=postgres")` or `die('No se ha podido conectar: ' . pg_last_error($conexion))`
- `$database_conexion` = "acti\_nacional"
- `$hostname_conexion` = "10.10.10.51"
- `$password_conexion` = "inec2012"
- `$username_conexion` = "postgres"

### Documentación de las variables

```
$conexion = pg_connect("host=localhost dbname=acti_nacional user=postgres  
password=postgres") or die('No se ha podido conectar: ' .  
pg_last_error($conexion))
```

Definición en la línea 10 del archivo conexion.php.

```
$database_conexion = "acti_nacional"
```

Definición en la línea 6 del archivo conexion.php.

```
$hostname_conexion = "10.10.10.51"
```

Definición en la línea 5 del archivo conexion.php.

```
$password_conexion = "inec2012"
```

Definición en la línea 8 del archivo conexion.php.

```
$username_conexion = "postgres"
```

Definición en la línea 7 del archivo conexion.php.

### ***Funciones***

- [if](#)((isset(\$\_SERVER['QUERY\_STRING']))&&(\$\_SERVER['QUERY\_STRING']!=""))
- [if](#)((isset(\$\_GET['doLogout']))&&(\$\_GET['doLogout']=="true")) [isAuthorized](#) (\$strUsers, \$strGroups, \$UserName, \$UserGroup)

### ***Documentación de las funciones***

```
if ((isset($_SERVER['QUERY_STRING']))&&($_SERVER['QUERY_STRING']!="")) if
((isset($_GET['doLogout']))&&($_GET['doLogout']=="true")) isAuthorized (
$strUsers, $strGroups, $UserName, $UserGroup)
```

Definición en la línea 27 del archivo Error.php.

### ***Variables***

- [\\$logoutAction](#) = \$\_SERVER['PHP\_SELF']."?doLogout=true"
- [if](#)(!isset(\$\_SESSION)) [\\$MM\\_authorizedUsers](#) = ""
- [\\$MM\\_donotCheckaccess](#) = "true"
- [\\$MM\\_restrictGoTo](#) = "../index.php"

### ***Documentación de las variables***

```
$logoutAction = $_SERVER['PHP_SELF']."?doLogout=true"
```

Definición en la línea 9 del archivo Error.php.

```
if (!isset($_SESSION)) $MM_authorizedUsers = ""
```

Definición en la línea 5 del archivo Error.php.

```
$MM_donotCheckaccess = "true"
```

Definición en la línea 6 del archivo Error.php.

```
$MM_restrictGoTo = "../index.php"
```

Definición en la línea 52 del archivo Error.php.

## **Referencia del Archivo: proyactinac/excelwriter.inc.php**

### ***Funciones***

- [close](#) ()
- [ExcelWriter](#) (\$file="")
- [GetFooter](#) ()
- [GetHeader](#) ()
- [open](#) (\$file)
- [writeCol](#) (\$value)
- [writeLine](#) (\$line\_arr)

- [writeRow \(\)](#)

### ***Documentación de las funciones***

#### **close ()**

Definición en la línea 75 del archivo excelwriter.inc.php.

#### **ExcelWriter ( \$file = "" )**

Definición en la línea 35 del archivo excelwriter.inc.php.

#### **GetFooter ()**

Definición en la línea 209 del archivo excelwriter.inc.php.

Referenciado por close().

#### **GetHeader ()**

Definición en la línea 98 del archivo excelwriter.inc.php.

Referenciado por open().

#### **open ( \$file )**

Definición en la línea 48 del archivo excelwriter.inc.php.

Referenciado por ExcelWriter().

#### **writeCol ( \$value )**

Definición en la línea 259 del archivo excelwriter.inc.php.

#### **writeLine ( \$line\_arr )**

Definición en la línea 219 del archivo excelwriter.inc.php.

#### **writeRow ()**

Definición en la línea 241 del archivo excelwriter.inc.php.

### ***Variables***

- [\\$error](#)
- [\\$newRow](#) =false
- [\\$state](#) ="CLOSED"
- Class [ExcelWriter](#)

### ***Documentación de las variables***

#### **\$error**

Definición en la línea 25 del archivo excelwriter.inc.php.

Referenciado por MySqlConnection\GetDatabaseList() y KT\_ErrorHandler().

#### **\$newRow =false**

Definición en la línea 27 del archivo excelwriter.inc.php.

#### **\$state ="CLOSED"**

Definición en la línea 26 del archivo excelwriter.inc.php.

## Class ExcelWriter

```
Valor inicial: {
    var $fp=null
```

Definición en la línea 22 del archivo excelwriter.inc.php.

## Referencia del Archivo:

### proyactinac/exportCambiosEncuesta.php

#### Variables

- `$excel` = new `ExcelWriter`("cambiosEncuesta.xls")
- `$mod_encuesta` = `pg_query($conexion, $query_mod_encuesta)` or `die('Error en la consulta SQL'. pg_last_error($conexion))`
- `$query_mod_encuesta` = `sprintf("SELECT cambios_admin_encuesta.id_usuario,usuario.nombres,usuario.apellidos,usuario.direccion_regional,cambios_admin_encuesta.cod_encuesta,encuesta.cod_direccion_regional,cambios_admin_encuesta.fecha_hora_ingreso,cambios_admin_encuesta.observacion FROM cambios_admin_encuesta, usuario, encuesta ".$where)`
- `$row_mod_encuesta` = `pg_fetch_assoc($mod_encuesta)`
- `if(isset($_REQUEST["regional"])&&$_REQUEST["regional"]!="")`
- `if((isset($_REQUEST["fecha_registro"])&&$_REQUEST["fecha_registro"]!="")&&(isset($_REQUEST["fecha_registro2"])&&$_REQUEST["fecha_registro2"]!="")) $select = ""`
- `$totalRows_mod_encuesta` = `pg_num_rows($mod_encuesta)`
- `if($excel==false) $where = "WHERE cambios_admin_encuesta.id_usuario = usuario.id_usuario AND cambios_admin_encuesta.cod_encuesta = encuesta.cod_encuesta"`

#### Documentación de las variables

`$excel = new ExcelWriter("cambiosEncuesta.xls")`

Definición en la línea 6 del archivo exportCambiosEncuesta.php.

`$mod_encuesta = pg_query($conexion, $query_mod_encuesta)` or `die('Error en la consulta SQL'. pg_last_error($conexion))`

Definición en la línea 57 del archivo exportCambiosEncuesta.php.

`$query_mod_encuesta = sprintf("SELECT cambios_admin_encuesta.id_usuario,usuario.nombres,usuario.apellidos,usuario.direccion_regional,cambios_admin_encuesta.cod_encuesta,encuesta.cod_direccion_regional,cambios_admin_encuesta.fecha_hora_ingreso,cambios_admin_encuesta.observacion FROM cambios_admin_encuesta, usuario, encuesta ".$where)`

Definición en la línea 56 del archivo exportCambiosEncuesta.php.

**\$row\_mod\_encuesta = pg\_fetch\_assoc(\$mod\_encuesta)**

Definición en la línea 58 del archivo exportCambiosEncuesta.php.

```
if      (isset($_REQUEST["regional"])&&$_REQUEST["regional"]!="")      if
((isset($_REQUEST["fecha_registro"])&&$_REQUEST["fecha_registro"]!="")&&(isset($_REQUEST["fecha_registro2"])&&$_REQUEST["fecha_registro2"]!=""))) $select
= ""
```

Definición en la línea 38 del archivo exportCambiosEncuesta.php.

**\$totalRows\_mod\_encuesta = pg\_num\_rows(\$mod\_encuesta)**

Definición en la línea 59 del archivo exportCambiosEncuesta.php.

```
if ($excel==false) $where = "WHERE cambios_admin_encuesta.id_usuario =
usuario.id_usuario      AND      cambios_admin_encuesta.cod_encuesta      =
encuesta.cod_encuesta"
```

Definición en la línea 12 del archivo exportCambiosEncuesta.php.

## Referencia del Archivo:

### proyactinac/exportCambiosMuestra.php

#### *Variables*

- `$excel` = new `ExcelWriter`("cambiosMuestra.xls")
- `$mod_muestra` = `pg_query`(\$conexion, \$query\_mod\_muestra) or die('Error en la consulta SQL'.  
`pg_last_error`(\$conexion))
- `$query_mod_muestra` = `sprintf`("SELECT  
cambios\_admin\_muestra.id\_usuario,usuario.nombres,usuario.apellidos,usuario.direccion\_regional,ca  
mbios\_admin\_muestra.cod\_empresa,muestra.cod\_direccion\_regional,cambios\_admin\_muestra.fecha  
\_hora\_ingreso,cambios\_admin\_muestra.observacion FROM cambios\_admin\_muestra, usuario,  
muestra ".\$where)
- `$row_mod_muestra` = `pg_fetch_assoc`(\$mod\_muestra)
- `if`(isset(\$\_REQUEST["regional"])&&\$\_REQUEST["regional"]!="")
- `if`((isset(\$\_REQUEST["fecha\_registro"])&&\$\_REQUEST["fecha\_registro"]!="")&&(isset(\$\_REQUEST["fecha\_registro2"])&&\$\_REQUEST["fecha\_registro2"]!=""))) `$select` = ""
- `$totalRows_mod_muestra` = `pg_num_rows`(\$mod\_muestra)
- `if`(\$excel==false) `$where` = "WHERE cambios\_admin\_muestra.id\_usuario = usuario.id\_usuario  
AND cambios\_admin\_muestra.cod\_empresa = muestra.cod\_empresa"

## Documentación de las variables

**\$excel = new [ExcelWriter](#)("cambiosMuestra.xls")**

Definición en la línea 6 del archivo exportCambiosMuestra.php.

**\$mod\_muestra = pg\_query(\$conexion, \$query\_mod\_muestra) or die('Error en la consulta SQL'. pg\_last\_error(\$conexion))**

Definición en la línea 57 del archivo exportCambiosMuestra.php.

**\$query\_mod\_muestra = sprintf("SELECT cambios\_admin\_muestra.id\_usuario,usuario.nombres,usuario.apellidos,usuario.direccion\_regional,cambios\_admin\_muestra.cod\_empresa,muestra.cod\_direccion\_regional,cambios\_admin\_muestra.fecha\_hora\_ingreso,cambios\_admin\_muestra.observacion FROM cambios\_admin\_muestra, usuario, muestra ".\$where)**

Definición en la línea 56 del archivo exportCambiosMuestra.php.

**\$row\_mod\_muestra = pg\_fetch\_assoc(\$mod\_muestra)**

Definición en la línea 58 del archivo exportCambiosMuestra.php.

**if (isset(\$\_REQUEST["regional"])&&\$\_REQUEST["regional"]!="") if ((isset(\$\_REQUEST["fecha\_registro"])&&\$\_REQUEST["fecha\_registro"]!="")&&(isset(\$\_REQUEST["fecha\_registro2"])&&\$\_REQUEST["fecha\_registro2"]!="))) \$select = ""**

Definición en la línea 38 del archivo exportCambiosMuestra.php.

**\$totalRows\_mod\_muestra = pg\_num\_rows(\$mod\_muestra)**

Definición en la línea 59 del archivo exportCambiosMuestra.php.

**if (\$excel==false) \$where = "WHERE cambios\_admin\_muestra.id\_usuario = usuario.id\_usuario AND cambios\_admin\_muestra.cod\_empresa = muestra.cod\_empresa"**

Definición en la línea 12 del archivo exportCambiosMuestra.php.

## Referencia del Archivo: proyactinac/exportDigitadores.php

### Variables

- `$digitadores = pg_query($conexion, $query_digitadores) or die('Error en la consulta SQL'. pg_last_error($conexion))`
  - `$excel = new ExcelWriter("digitadores.xls")`
  - `$query_digitadores = sprintf("SELECT u.nombres,u.apellidos,u.cedula,u.tipo,u.direccion_regional,u.carga_inicial,u.carga_final, COUNT(*) FROM usuario as u INNER JOIN encuesta as e ON u.id_usuario = e.id_usuario ".$where." GROUP BY u.nombres,u.apellidos,u.cedula,u.tipo,u.direccion_regional,u.carga_inicial,u.carga_final")`
  - `$row_digitadores = pg_fetch_assoc($digitadores)`
  - `if(isset($_REQUEST["regional"])&&$_REQUEST["regional"]!="")`
  - `if((isset($_REQUEST["fecha_registro"])&&$_REQUEST["fecha_registro"]!="")&&(isset($_REQUEST["fecha_registro2"])&&$_REQUEST["fecha_registro2"]!="")) $select = ""`
  - `$totalRows_digitadores = pg_num_rows($digitadores)`
- `if($excel==false) $where = " WHERE e.seccion >= 16 AND e.estado = 'C'"`

### Documentación de las variables

**`$digitadores = pg_query($conexion, $query_digitadores) or die('Error en la consulta SQL'. pg_last_error($conexion))`**

Definición en la línea 71 del archivo exportDigitadores.php.

**`$excel = new ExcelWriter("digitadores.xls")`**

Definición en la línea 6 del archivo exportDigitadores.php.

**`$query_digitadores = sprintf("SELECT u.nombres,u.apellidos,u.cedula,u.tipo,u.direccion_regional,u.carga_inicial,u.carga_final, COUNT(*) FROM usuario as u INNER JOIN encuesta as e ON u.id_usuario = e.id_usuario ".$where." GROUP BY u.nombres,u.apellidos,u.cedula,u.tipo,u.direccion_regional,u.carga_inicial,u.carga_final")`**

Definición en la línea 70 del archivo exportDigitadores.php.

**`$row_digitadores = pg_fetch_assoc($digitadores)`**

Definición en la línea 72 del archivo exportDigitadores.php.

```
if      (isset($_REQUEST["regional"])&&$_REQUEST["regional"]!="")      if
((isset($_REQUEST["fecha_registro"])&&$_REQUEST["fecha_registro"]!="")&&(isset($_REQUEST["fecha_registro2"])&&$_REQUEST["fecha_registro2"]!=""))) $select
= ""
```

Definición en la línea 38 del archivo exportDigitadores.php.

```
$totalRows_digitadores = pg_num_rows($digitadores)
```

Definición en la línea 73 del archivo exportDigitadores.php.

```
if ($excel==false) $where = " WHERE e.seccion >= 16 AND e.estado = 'C'"
```

Definición en la línea 12 del archivo exportDigitadores.php.

## Referencia del Archivo:

### proyactinac/exportDigitadoresInc.php

#### *Variables*

- `$digitadores` = `pg_query($conexion, $query_digitadores)` or `die('Error en la consulta SQL'. pg_last_error($conexion))`
- `$excel` = new `ExcelWriter("digitadoresInc.xls")`
- `$query_digitadores` = `sprintf("SELECT u.nombres,u.apellidos,u.cedula,u.tipo,u.direccion_regional,u.carga_inicial,u.carga_final, COUNT(*) FROM usuario as u INNER JOIN encuesta as e ON u.id_usuario = e.id_usuario ".$where." GROUP BY u.nombres,u.apellidos,u.cedula,u.tipo,u.direccion_regional,u.carga_inicial,u.carga_final")`
- `$row_digitadores` = `pg_fetch_assoc($digitadores)`
- `if(isset($_REQUEST["regional"])&&$_REQUEST["regional"]!="")`
- `if((isset($_REQUEST["fecha_registro"])&&$_REQUEST["fecha_registro"]!="")&&(isset($_REQUEST["fecha_registro2"])&&$_REQUEST["fecha_registro2"]!=""))) $select = ""`
- `$totalRows_digitadores` = `pg_num_rows($digitadores)`
- `if($excel==false) $where = " WHERE e.estado = 'T'"`

#### *Documentación de las variables*

```
$digitadores = pg_query($conexion, $query_digitadores) or die('Error en la consulta SQL'. pg_last_error($conexion))
```

Definición en la línea 71 del archivo exportDigitadoresInc.php.

```
$excel =new ExcelWriter("digitadoresInc.xls")
```

Definición en la línea 6 del archivo exportDigitadoresInc.php.

```
$query_digitadores = sprintf("SELECT u.nombres,u.apellidos,u.cedula,u.tipo,u.direccion_regional,u.carga_inicial,u.carga_final, COUNT(*) FROM usuario as u INNER JOIN encuesta as e ON u.id_usuario = e.id_usuario ".$where." GROUP BY u.nombres,u.apellidos,u.cedula,u.tipo,u.direccion_regional,u.carga_inicial,u.carga_final")
```

Definición en la línea 70 del archivo exportDigitadoresInc.php.

```
$row_digitadores = pg_fetch_assoc($digitadores)
```

Definición en la línea 72 del archivo exportDigitadoresInc.php.

```
if (isset($_REQUEST["regional"])&&$_REQUEST["regional"]!="") if ((isset($_REQUEST["fecha_registro"])&&$_REQUEST["fecha_registro"]!="")&&(isset($_REQUEST["fecha_registro2"])&&$_REQUEST["fecha_registro2"]!=""))) $select = ""
```

Definición en la línea 38 del archivo exportDigitadoresInc.php.

```
$totalRows_digitadores = pg_num_rows($digitadores)
```

Definición en la línea 73 del archivo exportDigitadoresInc.php.

```
if ($excel==false) $where = " WHERE e.estado = 'I'"
```

Definición en la línea 12 del archivo exportDigitadoresInc.php.

## Referencia del Archivo: [proyactinac/exportEncuesta.php](#)

- [\\$encuesta](#) = pg\_query(\$conexion, \$query\_encuesta) or die('Error en la consulta SQL'. pg\_last\_error(\$conexion))
- [\\$excel](#) =new [ExcelWriter](#)("encuestas.xls")
- [\\$query\\_encuesta](#) = sprintf("SELECT ".\$select." FROM encuesta ".\$where." AND estado = 'C'")
- [\\$row\\_encuesta](#) = pg\_fetch\_assoc(\$encuesta)
- [if](#)(isset(\$\_REQUEST["regional"])&&\$\_REQUEST["regional"]!="")
- [if](#)((isset(\$\_REQUEST["fecha\_registro"])&&\$\_REQUEST["fecha\_registro"]!="")&&(isset(\$\_REQUEST["fecha\_registro2"])&&\$\_REQUEST["fecha\_registro2"]!=""))) [\\$select](#) = ""
- [\\$totalRows\\_encuesta](#) = pg\_num\_rows(\$encuesta)
- [if](#)(\$excel==false) [\\$where](#) = " WHERE seccion >= 16"

### ***Documentación de las variables***

**\$encuesta = pg\_query(\$conexion, \$query\_encuesta) or die('Error en la consulta SQL'. pg\_last\_error(\$conexion))**

Definición en la línea 209 del archivo exportEncuesta.php.

**\$excel = new [ExcelWriter](#)("encuestas.xls")**

Definición en la línea 6 del archivo exportEncuesta.php.

**\$query\_encuesta = sprintf("SELECT ".\$select." FROM encuesta ".\$where." AND estado = 'C'")**

Definición en la línea 208 del archivo exportEncuesta.php.

**\$row\_encuesta = pg\_fetch\_assoc(\$encuesta)**

Definición en la línea 210 del archivo exportEncuesta.php.

**if (isset(\$\_REQUEST["regional"])&&\$\_REQUEST["regional"]!="") if  
((isset(\$\_REQUEST["fecha\_registro"])&&\$\_REQUEST["fecha\_registro"]!="")&&(isset(\$\_REQUEST["fecha\_registro2"])&&\$\_REQUEST["fecha\_registro2"]!="))) \$select  
= ""**

Definición en la línea 39 del archivo exportEncuesta.php.

**\$totalRows\_encuesta = pg\_num\_rows(\$encuesta)**

Definición en la línea 211 del archivo exportEncuesta.php.

**if (\$excel==false) \$where = " WHERE seccion >= 16"**

Definición en la línea 12 del archivo exportEncuesta.php.

### **Referencia del Archivo: [proyactinac/funciones.js](#)**

#### ***FUNCIONES***

- function [cargarPreg1\\_4Sec10](#) ()
- function [cargarProvincias](#) (codRegion)
- function [clickS10](#) ()

- function [clickS13](#) ()
- function [clickS6](#) ()
- function [clickS7](#) ()
- function [clickS8](#) ()
- function [conMayusculas](#) (field)
- function [consultarCobertura](#) ()
- function [consultarDigitador](#) ()
- function [consultarEmpresa](#) ()
- function [consultarEncuesta](#) ()
- function [controlarSeccion15](#) ()
- function [crearComboPreg4Sec8](#) (valorPreg4Sec8)
- function [guardarSeccion](#) ()
- function [limpiarPreg1Sec6](#) ()
- function [limpiarSec7](#) ()
- function [limpiarSec8](#) ()
- function [MM\\_validateForm](#) ()
- function [modificarEncuesta](#) ()
- function [modificarMuestra](#) ()
- function [numeros](#) (e)
- function [prueba](#) ()
- function [quitarPreg2Seccion3\\_1](#) ()
- function [quitarPregSeccion11](#) ()
- function [quitarPregSeccion12](#) ()
- function [quitarPregSeccion15](#) ()
- function [quitarPregSeccion4\\_1](#) ()
- function [quitarPregSeccion8](#) ()
- function [quitarSeccion13](#) ()
- function [reporteActualizacion](#) ()
- function [reporteDigitadores](#) ()
- function [soloNumero](#) (text)
- function [sumaPreg1Sec14](#) ()
- function [sumaPreg1Sec6](#) ()
- function [sumaPreg3Sec3](#) ()
- function [validarActAdminEnc](#) ()
- function [validarActAdminEncEstadoC](#) ()
- function [validarActAdminEncEstadoI](#) ()
- function [validarActAdminMuestra](#) ()
- function [validarActAdminMuestraEstadoA](#) ()
- function [validarActAdminMuestraEstadoI](#) ()
- function [validarPreg2Seccion3](#) ()
- function [validarPreg3Seccion4](#) ()
- function [validarRuc](#) ()
- function [validarSeccion1](#) ()
- function [validarSeccion10](#) (radioButton)
- function [validarSeccion11](#) (radioButton)
- function [validarSeccion12](#) (radioButton)
- function [validarSeccion13](#) (radioButton)
- function [validarSeccion14](#) (radioButton)
- function [validarSeccion15](#) (radioButton)
- function [validarSeccion2](#) (radioButton)
- function [validarSeccion3](#) (radioButton)
- function [validarSeccion4](#) (radioButton)
- function [validarSeccion5](#) (radioButton)
- function [validarSeccion6](#) (radioButton)
- function [validarSeccion7](#) (radioButton)
- function [validarSeccion8](#) (radioButton)
- function [validarSeccion9](#) (radioButton)

- function [validarTelefono](#) (txt)
- function [validarUso](#) ()
- function [verificarSec\\_3\\_4\\_5](#) ()

### ***Documentación de las funciones***

#### **function cargarPreg1\_4Sec10 ()**

Definición en la línea 2416 del archivo funciones.js.

#### **function cargarProvincias ( *codRegion* )**

Definición en la línea 2748 del archivo funciones.js.

#### **function clickS10 ()**

Definición en la línea 2977 del archivo funciones.js.

#### **function clickS13 ()**

Definición en la línea 2983 del archivo funciones.js.

#### **function clickS6 ()**

Definición en la línea 2956 del archivo funciones.js.

#### **function clickS7 ()**

Definición en la línea 2963 del archivo funciones.js.

#### **function clickS8 ()**

Definición en la línea 2970 del archivo funciones.js.

#### **function conMayusculas ( *field* )**

Definición en la línea 2695 del archivo funciones.js.

#### **function consultarCobertura ()**

Definición en la línea 3039 del archivo funciones.js.

#### **function consultarDigitador ()**

Definición en la línea 3023 del archivo funciones.js.

**function consultarEmpresa ()**

Definición en la línea 3007 del archivo funciones.js.

**function consultarEncuesta ()**

Definición en la línea 2990 del archivo funciones.js.

**function controlarSeccion15 ()**

Definición en la línea 2624 del archivo funciones.js.

**function crearComboPreg4Sec8 ( *valorPreg4Sec8* )**

Definición en la línea 1922 del archivo funciones.js.

**function guardarSeccion ()**

```
function cambiarColor(){ document.getElementById("usoI").style.color="#33CCCC";  
alert("color"); }
```

Definición en la línea 2757 del archivo funciones.js.

**function limpiarPreg1Sec6 ()**

```
function vaciarSeccion13() {  
if(document.getElementById("seccion13_1").style.display=='none'){ } var radios =  
['III_1_a','III_1_b','III_1_c','III_1_d','IV_1_a','IV_1_b','XI_1','XII_1'],bandPregSec13=0; for  
(var j in radios) { var total,valor; total=document.getElementsByName(radios[j]).length;  
for (var i=0;i<total;i++){ valor=document.getElementsByName(radios[j]).item(i).checked;  
if (valor){ if(i == 1) { bandPregSec13=bandPregSec13+1; } } } }  
if(bandPregSec13 == 8) { document.getElementById("XIII_1_a").checked = false;  
document.getElementById("XIII_1_b").checked = false;  
document.getElementById("XIII_1_c").checked = false;  
document.getElementById("XIII_1_d").checked = false;  
document.getElementById("XIII_1_e").checked = false;  
document.getElementById("XIII_1_f").checked = false;  
document.getElementById("XIII_1_g").checked = false;  
document.getElementById("XIII_1_h").checked = false;  
document.getElementById("XIII_1_i").checked = false;  
document.getElementById("XIII_1_j").checked = false;  
document.getElementById("XIII_1_k").checked = false;  
document.getElementById("seccion13_1").style.display = "none"; } else {  
bandPregSec13=0; document.getElementById("seccion13_1").style.display = "block"; } }
```

Definición en la línea 2869 del archivo funciones.js.

Referenciado por validarSeccion6().

**function limpiarSec7 ()**

Definición en la línea 2880 del archivo funciones.js.

Referenciado por validarSeccion7().

**function limpiarSec8 ()**

Definición en la línea 2900 del archivo funciones.js.  
Referenciado por validarSeccion8().

**function MM\_validateForm ()**

Definición en la línea 1739 del archivo funciones.js.

**function modificarEncuesta ()**

Definición en la línea 3064 del archivo funciones.js.

**function modificarMuestra ()**

Definición en la línea 3079 del archivo funciones.js.

**function numeros ( e)**

Definición en la línea 2698 del archivo funciones.js.

**function prueba ()**

Definición en la línea 3051 del archivo funciones.js.

**function quitarPreg2Seccion3\_1 ()**

Definición en la línea 1870 del archivo funciones.js.

**function quitarPregSeccion11 ()**

Definición en la línea 2453 del archivo funciones.js.

**function quitarPregSeccion12 ()**

Definición en la línea 2504 del archivo funciones.js.

**function quitarPregSeccion15 ()**

Definición en la línea 2635 del archivo funciones.js.

**function quitarPregSeccion4\_1 ()**

Definición en la línea 1965 del archivo funciones.js.

**function quitarPregSeccion8 ()**

Definición en la línea 2005 del archivo funciones.js.

**function quitarSeccion13 ()**

Definición en la línea 2538 del archivo funciones.js.

**function reporteActualizacion ()**

Definición en la línea 3096 del archivo funciones.js.

**function reporteDigitadores ()**

Definición en la línea 3032 del archivo funciones.js.

**function soloNumero ( text)**

Definición en la línea 2715 del archivo funciones.js.

**function sumaPreg1Sec14 ()**

Definición en la línea 2684 del archivo funciones.js.

**function sumaPreg1Sec6 ()**

Definición en la línea 2675 del archivo funciones.js.

**function sumaPreg3Sec3 ()**

Definición en la línea 2664 del archivo funciones.js.

Referenciado por quitarPreg2Seccion3\_1().

**function validarActAdminEnc ()**

Definición en la línea 27 del archivo funciones.js.

**function validarActAdminEncEstadoC ()**

Definición en la línea 58 del archivo funciones.js.

**function validarActAdminEncEstadoI ()**

Definición en la línea 76 del archivo funciones.js.

**function validarActAdminMuestra ()**

Definición en la línea 95 del archivo funciones.js.

**function validarActAdminMuestraEstadoA ()**

Definición en la línea 126 del archivo funciones.js.

**function validarActAdminMuestraEstadoI ()**

Definición en la línea 144 del archivo funciones.js.

**function validarPreg2Seccion3 ()**

Definición en la línea 1831 del archivo funciones.js.

**function validarPreg3Seccion4 ()**

Definición en la línea 1952 del archivo funciones.js.

**function validarRuc ()**

Definición en la línea 252 del archivo funciones.js.

**function validarSeccion1 ()**

Definición en la línea 203 del archivo funciones.js.

**function validarSeccion10 ( *radioButton* )**

Definición en la línea 1309 del archivo funciones.js.

**function validarSeccion11 ( *radioButton* )**

Definición en la línea 1364 del archivo funciones.js.

**function validarSeccion12 ( *radioButton* )**

Definición en la línea 1433 del archivo funciones.js.

**function validarSeccion13 ( *radioButton* )**

Definición en la línea 1502 del archivo funciones.js.

**function validarSeccion14 ( *radioButton* )**

Definición en la línea 1543 del archivo funciones.js.

**function validarSeccion15 ( *radioButton* )**

Definición en la línea 1636 del archivo funciones.js.

**function validarSeccion2 ( *radioButton* )**

Definición en la línea 262 del archivo funciones.js.

**function validarSeccion3 ( *radioButton* )**

Definición en la línea 367 del archivo funciones.js.

**function validarSeccion4 ( *radioButton* )**

Definición en la línea 581 del archivo funciones.js.

**function validarSeccion5 ( *radioButton* )**

Definición en la línea 716 del archivo funciones.js.

**function validarSeccion6 ( *radioButton* )**

Definición en la línea 927 del archivo funciones.js.

**function validarSeccion7 ( *radioButton* )**

Definición en la línea 1061 del archivo funciones.js.

**function validarSeccion8 ( *radioButton* )**

Definición en la línea 1115 del archivo funciones.js.

**function validarSeccion9 ( *radioButton* )**

Definición en la línea 1220 del archivo funciones.js.

**function validarTelefono ( *txt* )**

Definición en la línea 242 del archivo funciones.js.

**function validarUso ( )**

Definición en la línea 167 del archivo funciones.js.

**function verificarSec\_3\_4\_5 ( )**

Definición en la línea 2772 del archivo funciones.js.

Referenciado por validarSeccion7() y validarSeccion8().

•

## VARIABLES

- var [bandInn](#) = 0
- var [bandPreg2Sec3](#) = 0
- var [bandPregSec4](#) = 0
- var [bandS7S8](#) = 0

- var [codRegion](#)
- var [errors1](#) = "
- var [errors10](#) = "
- var [errors11](#) = "
- var [errors12](#) = "
- var [errors13](#) = "
- var [errors14](#) = "
- var [errors15](#) = "
- var [errors1a](#) = "
- var [errors2](#) = "
- var [errors3](#) = "
- var [errors4](#) = "
- var [errors5](#) = "
- var [errors6](#) = "
- var [errors7](#) = "
- var [errors8](#) = "
- var [errors9](#) = "
- var [errorsAd1](#) = "
- var [errorsAd2](#) = "
- var [radioButton](#)

### ***Documentación de las variables***

#### **var bandInn = 0**

Definición en la línea 23 del archivo funciones.js.  
Referenciado por validarSeccion7() y validarSeccion8().

#### **var bandPreg2Sec3 = 0**

Definición en la línea 21 del archivo funciones.js.  
Referenciado por quitarPreg2Seccion3\_1().

#### **var bandPregSec4 = 0**

Definición en la línea 22 del archivo funciones.js.  
Referenciado por quitarPregSeccion4\_1().

#### **var bandS7S8 = 0**

Definición en la línea 24 del archivo funciones.js.  
Referenciado por validarSeccion7() y validarSeccion8().

#### **var codRegion**

```
function letras(e) { key = e.keyCode || e.which; tecla2 = String.fromCharCode(key).toUpperCase(); l2 = " áéíóúabcdefghijklmnopqrstuvwxy"; especiales = [8, 37, 39, 46, 64, 09]; tecla_especial = false for(var i in especiales) { if(key == especiales[i]){ tecla_especial = true; break; } } if(l.indexOf(tecla2) == -1 && !tecla_especial) return false; } function limpiar() { var val = document.getElementById("input").value; var tam = val.length; for(i = 0; i < tam; i++) { if(!isNaN(val[i])) document.getElementById("input").value = ""; } }
```

Definición en la línea 2747 del archivo funciones.js.

#### **var errors1 = "**

Definición en la línea 5 del archivo funciones.js.  
Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(), validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),

validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors10 = "**

Definición en la línea 15 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors11 = "**

Definición en la línea 16 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors12 = "**

Definición en la línea 17 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors13 = "**

Definición en la línea 18 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors14 = "**

Definición en la línea 19 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors15 = "**

Definición en la línea 20 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors1a = "**

Definición en la línea 6 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors2 = "**

Definición en la línea 7 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),

validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors3 = "**

Definición en la línea 8 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors4 = "**

Definición en la línea 9 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors5 = "**

Definición en la línea 10 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors6 = "**

Definición en la línea 11 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors7 = "**

Definición en la línea 12 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors8 = "**

Definición en la línea 13 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errors9 = "**

Definición en la línea 14 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarSeccion1(), validarSeccion10(), validarSeccion11(),  
validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(),  
validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(),  
validarSeccion8(), validarSeccion9() y validarUso().

**var errorsAd1 = "**

Definición en la línea 3 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarActAdminEnc(), validarSeccion1(),  
validarSeccion10(), validarSeccion11(), validarSeccion12(), validarSeccion13(), validarSeccion14(),

validarSeccion15(), validarSeccion2(), validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(), validarSeccion8(), validarSeccion9() y validarUso().

**var errorsAd2 = "**

Definición en la línea 4 del archivo funciones.js.

Referenciado por MM\_validateForm(), validarActAdminMuestra(), validarSeccion1(), validarSeccion10(), validarSeccion11(), validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(), validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(), validarSeccion8(), validarSeccion9() y validarUso().

**var radioButton**

Definición en la línea 2 del archivo funciones.js.

Referenciado por validarSeccion10(), validarSeccion11(), validarSeccion12(), validarSeccion13(), validarSeccion14(), validarSeccion15(), validarSeccion2(), validarSeccion3(), validarSeccion4(), validarSeccion5(), validarSeccion6(), validarSeccion7(), validarSeccion8() y validarSeccion9().

## Referencia del Archivo: proyactinac/index.php

### FUNCIONES

- [if\(isset\(\\$\\_REQUEST\["band"\]\)\) is\\_chrome \(\)](#)

#### *Documentación de las funciones*

**[if \(isset\(\\$\\_REQUEST\["band"\]\)\) is\\_chrome \(\)](#)**

Definición en la línea 195 del archivo index.php.

### VARIABLES

- [if\(!function\\_exists\("GetSQLValueString"\)\)](#)
- [if\(!isset\(\\$\\_SESSION\)\) \\$loginFormAction = \\$\\_SERVER\['PHP\\_SELF'\]](#)
- [if\(isset\(\\$\\_GET\['accesscheck'\]\)\) if \(isset\(\\$\\_POST\['cedula'\]\)\) {](#)

#### *Documentación de las variables*

**[if \(!function\\_exists\("GetSQLValueString"\)\) if \(!isset\(\\$\\_SESSION\)\) \\$loginFormAction = \\$\\_SERVER\['PHP\\_SELF'\]](#)**

Definición en la línea 40 del archivo index.php.

**if {**

Definición en la línea 45 del archivo index.php.

## Referencia del Archivo: proyactinac/index\_respaldo.js

### FUNCIONES

- `if((isset($_SERVER['QUERY_STRING']))&&($_SERVER['QUERY_STRING']!=""))`
- `if((isset($_GET['doLogout']))&&($_GET['doLogout']=="true")) isAuthorized ($strUsers, $strGroups, $UserName, $UserGroup)`

#### Documentación de las funciones

**if** ((isset(\$\_SERVER['QUERY\_STRING']))&&(\$\_SERVER['QUERY\_STRING']!="")) **if** ((isset(\$\_GET['doLogout']))&&(\$\_GET['doLogout']=="true")) **isAuthorized** (***\$strUsers, \$strGroups, \$UserName, \$UserGroup***)

Definición en la línea 28 del archivo index\_respaldo.php.

### VARIABLES

- `\$cod\_ciiu = pg_query($conexion, $query_cod_ciiu) or die('Error en la consulta SQL'. pg_last_error($conexion))`
- `\$cod\_cpc = pg_query($conexion, $query_cod_cpc) or die('Error en la consulta SQL'. pg_last_error($conexion))`
- `if(!(isset($_SESSION['MM_Username']))&&(isAuthorized("", $MM_authorizedUsers, $_SESSION['MM_Username'], $_SESSION['MM_UserGroup'])))`
- `if(!function_exists("GetSQLValueString")) \$colname\_usuario = "-1"`
- `if(isset($_REQUEST["id"])) \$cont = 2`
- `\$encuesta\_i = pg_query($conexion, $query_encuesta) or die('Error en la consulta SQL'. pg_last_error($conexion))`
- `\$logoutAction = $_SERVER['PHP_SELF']."?doLogout=true"`
- `if(!isset($_SESSION)) \$MM\_authorizedUsers = ""`
- `\$MM\_donotCheckaccess = "true"`
- `\$MM\_restrictGoTo = "../index.php"`
- `\$preguntas = pg_query($conexion, $query_preguntas) or die('Error en la consulta SQL'. pg_last_error($conexion))`
- `\$query\_cod\_ciiu = "SELECT DISTINCT * FROM ciiu ORDER BY cod_ciiu ASC"`
- `\$query\_cod\_cpc = "SELECT DISTINCT * FROM cpc ORDER BY cod_cpc ASC"`
- `if($totalRows_preguntas >`
- `0) if($totalRows_secciones >`
- `0) if($row_usuario["tipo"]=="Analista")`
- `if($row_usuario["tipo"]=="Encuestador")`
- `if(isset($_REQUEST["id"])) \$query\_encuesta = "SELECT * FROM encuesta WHERE id_usuario = ".$row_usuario["id_usuario"]." AND seccion < 16 order by id_encuesta ASC LIMIT 1"`
- `\$query\_preguntas = "SELECT * FROM pregunta"`
- `if($totalRows_cod_cpc > 0) \$query\_regional = "SELECT DISTINCT * FROM regional WHERE estado = 'A' ORDER BY nom_regional ASC"`
- `\$query\_secciones = "SELECT * FROM seccion"`

- `if(isset($_SESSION['MM_Username'])) $query_usuario = sprintf("SELECT * FROM usuario WHERE cedula = %s ORDER BY apellidos ASC", GetSQLValueString($colname_usuario, "text"))`
- `$query_usuario_enc = "SELECT * FROM usuario WHERE tipo = 'Encuestador' AND estado = 'A' ORDER BY apellidos ASC"`
- `$reg_encuesta = pg_fetch_assoc($encuesta_i)`
- `$regional = pg_query($conexion, $query_regional) or die('Error en la consulta SQL'. pg_last_error($conexion))`
- `$row_cod_ciiu = pg_fetch_assoc($cod_ciiu)`
- `$row_cod_cpc = pg_fetch_assoc($cod_cpc)`
- `$row_preguntas = pg_fetch_assoc($preguntas)`
- `$row_regional = pg_fetch_assoc($regional)`
- `$row_secciones = pg_fetch_assoc($secciones)`
- `$row_usuario = pg_fetch_assoc($usuario)`
- `$row_usuario_enc = pg_fetch_assoc($usuario_enc)`
- `$secciones = pg_query($conexion, $query_secciones) or die('Error en la consulta SQL'. pg_last_error($conexion))`
- `$total_Reg_encuesta = pg_num_rows($encuesta_i)`
- `if(isset($_REQUEST["codProvincia"]))`
- `if(isset($_REQUEST["id"])&&!isset($_REQUEST["codRegion"])&&!isset($_REQUEST["codProvincia"])) $totalRows_canton = 0`
- `if($totalRows_usuario_enc > 0) $totalRows_cod_ciiu = 0`
- `if($totalRows_cod_ciiu > 0) $totalRows_cod_cpc = 0`
- `if(isset($_REQUEST["codCanton"]))`
- `if(isset($_REQUEST["id"])&&!isset($_REQUEST["codRegion"])&&!isset($_REQUEST["codProvincia"])&&!isset($_REQUEST["codCanton"])) $totalRows_parroquia = 0`
- `$totalRows_preguntas = pg_num_rows($preguntas)`
- `if(isset($_REQUEST["codRegion"]))`
- `if(isset($_REQUEST["id"])&&!isset($_REQUEST["codRegion"])) $totalRows_provincia = 0`
- `$totalRows_regional = pg_num_rows($regional)`
- `$totalRows_secciones = pg_num_rows($secciones)`
- `$totalRows_usuario = pg_num_rows($usuario)`
- `$totalRows_usuario_enc = 0`
- `$usuario = pg_query($conexion, $query_usuario) or die('Error en la consulta SQL'. pg_last_error($conexion))`
- `$usuario_enc = pg_query($conexion, $query_usuario_enc) or die('Error en la consulta SQL'. pg_last_error($conexion))`
- `$val_carga_fin = $row_usuario["carga_final"]`
- `$val_carga_ini = $row_usuario["carga_inicial"]`
- `if($whereProvincia!="") if($totalRows_canton > 0) $whereCanton = ""`
- `if($whereRegion!="") if($totalRows_provincia > 0) $whereProvincia = ""`
- `if($totalRows_regional > 0) $whereRegion = ""`
- `if($total_Reg_encuesta==0) else`
- `if($whereCanton!="") if($totalRows_parroquia > 0) if (isset($_REQUEST["buscar_cod_encuesta"])&&$_REQUEST["buscar_cod_encuesta"]!="")`
- `if ($row_encuesta["seccion"]!="")`
- `if (isset($_REQUEST["id"]))`
- `if($totalRows_secciones >`
- `0) if((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15")) if ((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15"))`

### ***Documentación de las variables***

**\$cod\_ciiu = pg\_query(\$conexion, \$query\_cod\_ciiu) or die('Error en la consulta SQL'. pg\_last\_error(\$conexion))**

Definición en la línea 194 del archivo index\_respaldo.php.

```
$cod_cpc = pg_query($conexion, $query_cod_cpc) or die('Error en la consulta SQL'.  
pg_last_error($conexion))
```

Definición en la línea 215 del archivo index\_respaldo.php.

```
if (!((isset($_SESSION['MM_Username']))&&(isAuthorized("",  
$MM_authorizedUsers, $_SESSION['MM_Username'],  
$_SESSION['MM_UserGroup'])))) if (!function_exists("GetSQLValueString"))  
$colname_usuario = "-1"
```

Definición en la línea 98 del archivo index\_respaldo.php.

```
if (isset($_REQUEST["id"])) $cont = 2
```

Definición en la línea 668 del archivo index\_respaldo.php.

```
$encuesta_i = pg_query($conexion, $query_encuesta) or die('Error en la consulta  
SQL'. pg_last_error($conexion))
```

Definición en la línea 167 del archivo index\_respaldo.php.

```
$logoutAction = $_SERVER['PHP_SELF']."?doLogout=true"
```

Definición en la línea 10 del archivo index\_respaldo.php.

```
if (!isset($_SESSION)) $MM_authorizedUsers = ""
```

Definición en la línea 6 del archivo index\_respaldo.php.

```
$MM_donotCheckaccess = "true"
```

Definición en la línea 7 del archivo index\_respaldo.php.

```
$MM_restrictGoTo = "../index.php"
```

Definición en la línea 53 del archivo index\_respaldo.php.

```
$preguntas = pg_query($conexion, $query_preguntas) or die('Error en la consulta  
SQL'. pg_last_error($conexion))
```

Definición en la línea 119 del archivo index\_respaldo.php.

```
$query_cod_ciiu = "SELECT DISTINCT * FROM ciiu ORDER BY cod_ciiu ASC"
```

Definición en la línea 193 del archivo index\_respaldo.php.

```
$query_cod_cpc = "SELECT DISTINCT * FROM cpc ORDER BY cod_cpc ASC"
```

Definición en la línea 214 del archivo index\_respaldo.php.

```
if ($totalRows_preguntas > 0) if ($totalRows_secciones > 0) if ($row_usuario["tipo"]=="Analista") if ($row_usuario["tipo"]=="Encuestador") if (isset($_REQUEST["id"])) $query_encuesta = "SELECT * FROM encuesta WHERE id_usuario = ".$row_usuario['id_usuario']." AND seccion < 16 order by id_encuesta ASC LIMIT 1"
```

```
if(isset($_REQUEST["id"])){ consultas de encuestas por código $query_cod_encuesta = "SELECT * FROM encuesta WHERE cod_encuesta = 52"; $cod_encuesta = pg_query($conexion, $query_cod_encuesta) or die('Error en la consulta SQL. Código de encuesta'. pg_last_error($conexion)); $row_cod_encuesta = pg_fetch_assoc($cod_encuesta); $totalRows_cod_encuesta = pg_num_rows($cod_encuesta); echo $row_cod_encuesta[0]["cod_encuesta"]; }
```

Definición en la línea 166 del archivo index\_respaldo.php.

```
$query_preguntas = "SELECT * FROM pregunta"
```

Definición en la línea 118 del archivo index\_respaldo.php.

```
if ($totalRows_cod_cpc > 0) $query_regional = "SELECT DISTINCT * FROM regional WHERE estado = 'A' ORDER BY nom_regional ASC"
```

```
$whereCpc = ""; if(isset($_REQUEST["codCpc"])){ $whereCiiu = " WHERE cod_cpc = ".$_REQUEST["codCpc"].""; } if(isset($_REQUEST["id"]) && !isset($_REQUEST["codCpc"])){ $whereCiiu = " WHERE cod_cpc = ".$_row_encuesta['ii_cpc_8_a'].""; } if(isset($_REQUEST["cod_actividad_economica"])) { $query_act_economica = "SELECT actividad_eco FROM ciiu WHERE cod_ciiu = ".$_REQUEST["cod_actividad_economica"].""; $act_economica = pg_query($conexion, $query_act_economica) or die('Error en la consulta SQL'. pg_last_error($conexion)); $reg_act_economica = pg_fetch_assoc($act_economica); $total_Reg_act_economica = pg_num_rows($act_economica); }
```

Definición en la línea 247 del archivo index\_respaldo.php.

```
$query_secciones = "SELECT * FROM seccion"
```

Definición en la línea 112 del archivo index\_respaldo.php.

```
if (isset($_SESSION['MM_Username'])) $query_usuario = sprintf("SELECT * FROM usuario WHERE cedula = %0s ORDER BY apellidos ASC", GetSQLValueString($colname_usuario, "text"))
```

Definición en la línea 103 del archivo index\_respaldo.php.

```
$query_usuario_enc = "SELECT * FROM usuario WHERE tipo = 'Encuestador' AND estado = 'A' ORDER BY apellidos ASC"
```

Definición en la línea 179 del archivo index\_respaldo.php.

```
$reg_encuesta = pg_fetch_assoc($encuesta_i)
```

Definición en la línea 168 del archivo index\_respaldo.php.

```
$regional = pg_query($conexion, $query_regional) or die('Error en la consulta SQL'. pg_last_error($conexion))
```

Definición en la línea 248 del archivo index\_respaldo.php.

```
$row_cod_ciiu = pg_fetch_assoc($cod_ciiu)
```

Definición en la línea 195 del archivo index\_respaldo.php.

```
$row_cod_cpc = pg_fetch_assoc($cod_cpc)
```

Definición en la línea 216 del archivo index\_respaldo.php.

```
$row_preguntas = pg_fetch_assoc($preguntas)
```

Definición en la línea 120 del archivo index\_respaldo.php.

```
$row_regional = pg_fetch_assoc($regional)
```

Definición en la línea 249 del archivo index\_respaldo.php.

```
$row_secciones = pg_fetch_assoc($secciones)
```

Definición en la línea 114 del archivo index\_respaldo.php.

```
$row_usuario = pg_fetch_assoc($usuario)
```

Definición en la línea 105 del archivo index\_respaldo.php.

```
$row_usuario_enc = pg_fetch_assoc($usuario_enc)
```

Definición en la línea 181 del archivo index\_respaldo.php.

```
$secciones = pg_query($conexion, $query_secciones) or die('Error en la consulta SQL'. pg_last_error($conexion))
```

Definición en la línea 113 del archivo index\_respaldo.php.

**\$total\_Reg\_encuesta = pg\_num\_rows(\$encuesta\_i)**

Definición en la línea 169 del archivo index\_respaldo.php.

```
if (isset($_REQUEST["codProvincia"])) if
(
isset($_REQUEST["id"])&&isset($_REQUEST["codRegion"])&&isset($_REQUEST['
codProvincia'])) $totalRows_canton = 0
```

Definición en la línea 288 del archivo index\_respaldo.php.

**\$totalRows\_cod\_ciiu = 0**

Definición en la línea 192 del archivo index\_respaldo.php.

**\$totalRows\_cod\_cpc = 0**

```
$whereCiiu = ""; if(isset($_REQUEST["codCiiu"])){ $whereCiiu = " WHERE cod_ciiu =
".$_REQUEST['codCiiu'].""; } if(isset($_REQUEST["id"]) &&
!isset($_REQUEST["codCiiu"])){ $whereCiiu = " WHERE cod_ciiu =
".$_row_encuesta['cod_actividad_economica'].""; } $totalRows_cod_cpc = 0; if($whereCiiu
!= ""){
```

Definición en la línea 213 del archivo index\_respaldo.php.

```
if (isset($_REQUEST["codCanton"])) if
(
isset($_REQUEST["id"])&&isset($_REQUEST['codRegion'])&&isset($_REQUEST['c
odProvincia'])&&isset($_REQUEST['codCanton'])) $totalRows_parroquia = 0
```

Definición en la línea 311 del archivo index\_respaldo.php.

**\$totalRows\_preguntas = pg\_num\_rows(\$preguntas)**

Definición en la línea 121 del archivo index\_respaldo.php.

```
if (isset($_REQUEST["codRegion"])) if
(
isset($_REQUEST["id"])&&isset($_REQUEST["codRegion"]))
$totalRows_provincia = 0
```

Definición en la línea 265 del archivo index\_respaldo.php.

**\$totalRows\_regional = pg\_num\_rows(\$regional)**

Definición en la línea 250 del archivo index\_respaldo.php.

**\$totalRows\_secciones = pg\_num\_rows(\$secciones)**

Definición en la línea 115 del archivo index\_respaldo.php.

**`$totalRows_usuario = pg_num_rows($usuario)`**

Definición en la línea 106 del archivo index\_respaldo.php.

**`$totalRows_usuario_enc = 0`**

Definición en la línea 178 del archivo index\_respaldo.php.

**`$usuario = pg_query($conexion, $query_usuario) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`**

Definición en la línea 104 del archivo index\_respaldo.php.

**`$usuario_enc = pg_query($conexion, $query_usuario_enc) or die('Error en la  
consulta SQL'. pg_last_error($conexion))`**

Definición en la línea 180 del archivo index\_respaldo.php.

**`$val_carga_fin = $row_usuario["carga_final"]`**

Definición en la línea 109 del archivo index\_respaldo.php.

**`$val_carga_ini = $row_usuario["carga_inicial"]`**

Definición en la línea 108 del archivo index\_respaldo.php.

**`if ($whereProvincia!="") if ($totalRows_canton > 0) $whereCanton = ""`**

Definición en la línea 304 del archivo index\_respaldo.php.

**`if ($whereRegion!="") if ($totalRows_provincia > 0) $whereProvincia = ""`**

Definición en la línea 281 del archivo index\_respaldo.php.

**`if ($totalRows_regional > 0) $whereRegion = ""`**

Definición en la línea 258 del archivo index\_respaldo.php.

**else**

```
Valor inicial:{  
    $bandRegEncuesta=0
```

Definición en la línea 172 del archivo index\_respaldo.php.

```

if ($whereCanton!="") if ($totalRows_parroquia > 0)
if(isset($_REQUEST["buscar_cod_encuesta"])&&$_REQUEST["buscar_cod_encues
ta"]!="")
    if(!isset($_REQUEST["buscar_cod_encuesta"]) &&
    !$_REQUEST["buscar_cod_encuesta"] != "") {
    $singCodEncuesta=$_REQUEST["buscar_cod_encuesta"];
    $query_cod_encuesta=sprintf("SELECT * FROM encuesta WHERE
cod_encuesta=%s",GetSQLValueString($singCodEncuesta, "text"));
    $cod_encuesta = pg_query($conexion, $query_cod_encuesta) or die('No se ha podido
establecer la conexión'); $row_cod_encuesta = pg_fetch_assoc($cod_encuesta);
    $totalRows_cod_encuesta = pg_num_rows($cod_encuesta);
    }

```

Definición en la línea 327 del archivo index\_respaldo.php.

```

if($row_encuesta["seccion"]!= "")

```

Definición en la línea 542 del archivo index\_respaldo.php.

```

if ($totalRows_parroquia > 0) if (isset($_REQUEST["id"])) if
(isset($_REQUEST["id"])) if ($totalRows_provincia > 0) if (isset($_REQUEST["id"]))
if (isset($_REQUEST["id"])) if (isset($_REQUEST["id"])) if ($totalRows_canton > 0)
if (isset($_REQUEST["id"])) if (isset($_REQUEST["id"])) if ($totalRows_usuario_enc
> 0) if (isset($_REQUEST["id"])) if(isset($_REQUEST["id"]))

```

Definición en la línea 599 del archivo index\_respaldo.php.

```

if ($totalRows_secciones > 0) if
((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15"))
if((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15"))

```

Definición en la línea 3560 del archivo index\_respaldo.php.

## Referencia del Archivo: proyactinac/indexEnc.php

### FUNCIONES

- [if](#)((isset(\$\_SERVER['QUERY\_STRING'])&&(\$\_SERVER['QUERY\_STRING']!="")))
- [if](#)((isset(\$\_GET['doLogout'])&&(\$\_GET['doLogout']=="true"))) [isAuthorized](#) (\$strUsers, \$strGroups, \$UserName, \$UserGroup)

## Documentación de las funciones

**if** ((isset(\$\_SERVER['QUERY\_STRING']))&&(\$\_SERVER['QUERY\_STRING']!="")) **if** ((isset(\$\_GET['doLogout']))&&(\$\_GET['doLogout']="true")) **isAuthorized** (***\$strUsers, \$strGroups, \$UserName, \$UserGroup***)

Definición en la línea 28 del archivo index\_respaldo.php.

## VARIABLES

- `$cod_ciiu` = pg\_query(\$conexion, \$query\_cod\_ciiu) or die('Error en la consulta SQL'. pg\_last\_error(\$conexion))
- `$cod_cpc` = pg\_query(\$conexion, \$query\_cod\_cpc) or die('Error en la consulta SQL'. pg\_last\_error(\$conexion))
- `if` (!(isset(\$\_SESSION['MM\_Username']))&&(isAuthorized("", \$MM\_authorizedUsers, \$\_SESSION['MM\_Username'], \$\_SESSION['MM\_UserGroup'])))
- `if` (!function\_exists("GetSQLValueString")) `$colname_usuario` = "-1"
- `if` (isset(\$\_REQUEST["id"])) `if` (isset(\$\_REQUEST["id"]))&&\$row\_encuesta["seccion"] >= 16 &&\$row\_encuesta["estado"] == 'C') `$cont` = 2
- `$encuesta_i` = pg\_query(\$conexion, \$query\_encuesta) or die('Error en la consulta SQL'. pg\_last\_error(\$conexion))
- `$logoutAction` = \$\_SERVER['PHP\_SELF']."?doLogout=true"
- `if` (isset(\$\_SESSION)) `$MM_authorizedUsers` = ""
- `$MM_donotCheckaccess` = "true"
- `$MM_restrictGoTo` = "../index.php"
- `$preguntas` = pg\_query(\$conexion, \$query\_preguntas) or die('Error en la consulta SQL'. pg\_last\_error(\$conexion))
- `$query_cod_ciiu` = "SELECT DISTINCT \* FROM ciiu ORDER BY cod\_ciiu ASC"
- `$query_cod_cpc` = "SELECT DISTINCT \* FROM cpc ORDER BY cod\_cpc ASC"
- `if` (\$totalRows\_preguntas > 0) `if` (\$totalRows\_secciones > 0) `if` (isset(\$\_REQUEST["id"])) `$query_encuesta` = "SELECT \* FROM encuesta\_enc WHERE id\_usuario = ".\$\_row\_usuario['id\_usuario']." AND seccion < 16 order by id\_encuesta\_enc ASC LIMIT 1"
- `$query_preguntas` = "SELECT \* FROM pregunta"
- `if` (\$totalRows\_cod\_cpc > 0) `$query_regional` = "SELECT DISTINCT \* FROM regional WHERE estado = 'A' ORDER BY nom\_regional ASC"
- `$query_secciones` = "SELECT \* FROM seccion"
- `if` (isset(\$\_SESSION['MM\_Username'])) `$query_usuario` = sprintf("SELECT \* FROM usuario WHERE cedula = %s ORDER BY apellidos ASC", GetSQLValueString(\$colname\_usuario, "text"))
- `$query_usuario_enc` = "SELECT \* FROM usuario WHERE tipo = 'Encuestador' AND estado = 'A' ORDER BY apellidos ASC"
- `$reg_encuesta` = pg\_fetch\_assoc(\$encuesta\_i)
- `$regional` = pg\_query(\$conexion, \$query\_regional) or die('Error en la consulta SQL'. pg\_last\_error(\$conexion))
- `$row_cod_ciiu` = pg\_fetch\_assoc(\$cod\_ciiu)
- `$row_cod_cpc` = pg\_fetch\_assoc(\$cod\_cpc)
- `$row_preguntas` = pg\_fetch\_assoc(\$preguntas)
- `$row_regional` = pg\_fetch\_assoc(\$regional)
- `$row_secciones` = pg\_fetch\_assoc(\$secciones)
- `$row_usuario` = pg\_fetch\_assoc(\$usuario)

- `$row_usuario_enc = pg_fetch_assoc($usuario_enc)`
- `$secciones = pg_query($conexion, $query_secciones) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`
- `$total_Reg_encuesta = pg_num_rows($encuesta_i)`
- `if(isset($_REQUEST["codProvincia"]))`
- `if(isset($_REQUEST["id"])&&!isset($_REQUEST["codRegion"])&&!isset($_REQUEST["codProvincia"])) $totalRows_canton = 0`
- `if($totalRows_usuario_enc > 0) $totalRows_cod_ciiu = 0`
- `if($totalRows_cod_ciiu > 0) $totalRows_cod_cpc = 0`
- `if(isset($_REQUEST["codCanton"]))`
- `if(isset($_REQUEST["id"])&&!isset($_REQUEST["codRegion"])&&!isset($_REQUEST["codProvincia"])&&!isset($_REQUEST["codCanton"])) $totalRows_parroquia = 0`
- `$totalRows_preguntas = pg_num_rows($preguntas)`
- `if(isset($_REQUEST["codRegion"]))`
- `if(isset($_REQUEST["id"])&&!isset($_REQUEST["codRegion"])) $totalRows_provincia = 0`
- `$totalRows_regional = pg_num_rows($regional)`
- `$totalRows_secciones = pg_num_rows($secciones)`
- `$totalRows_usuario = pg_num_rows($usuario)`
- `$totalRows_usuario_enc = 0`
- `$usuario = pg_query($conexion, $query_usuario) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`
- `$usuario_enc = pg_query($conexion, $query_usuario_enc) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`
- `$val_carga_fin = $row_usuario["carga_final"]`
- `$val_carga_ini = $row_usuario["carga_inicial"]`
- `if($whereProvincia!="") if($totalRows_canton > 0) $whereCanton = ""`
- `if($whereRegion!="") if($totalRows_provincia > 0) $whereProvincia = ""`
- `if($totalRows_regional > 0) $whereRegion = ""`
- `if($total_Reg_encuesta==0) else`
- `if($whereCanton!="") if($totalRows_parroquia > 0) if  
(isset($_REQUEST["buscar_cod_encuesta"])&&$_REQUEST["buscar_cod_encuesta"]!="")`
- `if ($row_encuesta["seccion"]!=" ")`
- `if(isset($_REQUEST["id"])) if (isset($_REQUEST["id"]))`
- `if($totalRows_secciones >`
- `0) if((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15")) if  
((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15"))`

### ***Documentación de las variables***

**`$cod_ciiu = pg_query($conexion, $query_cod_ciiu) or die('Error en la consulta SQL'. pg_last_error($conexion))`**

Definición en la línea 186 del archivo indexEnc.php.

**`$cod_cpc = pg_query($conexion, $query_cod_cpc) or die('Error en la consulta SQL'. pg_last_error($conexion))`**

Definición en la línea 207 del archivo indexEnc.php.

**`if (!((isset($_SESSION['MM_Username']))&&(isAuthorized("", $MM_authorizedUsers, $_SESSION['MM_Username'], $_SESSION['MM_UserGroup'])))) if (!function_exists("GetSQLValueString")) $colname_usuario = "-1"`**

Definición en la línea 98 del archivo indexEnc.php.

```
if (isset($_REQUEST["id"])) if
( (isset($_REQUEST["id"])&&$row_encuesta["seccion"] >=16
&&$row_encuesta["estado"]== 'C') $cont = 2
```

Definición en la línea 633 del archivo indexEnc.php.

```
$encuesta_i = pg_query($conexion, $query_encuesta) or die('Error en la consulta
SQL'. pg_last_error($conexion))
```

Definición en la línea 159 del archivo indexEnc.php.

```
$logoutAction = $_SERVER['PHP_SELF']."?doLogout=true"
```

Definición en la línea 10 del archivo indexEnc.php.

```
if (!isset($_SESSION)) $MM_authorizedUsers = ""
```

Definición en la línea 6 del archivo indexEnc.php.

```
$MM_donotCheckaccess = "true"
```

Definición en la línea 7 del archivo indexEnc.php.

```
$MM_restrictGoTo = "../index.php"
```

Definición en la línea 53 del archivo indexEnc.php.

```
$preguntas = pg_query($conexion, $query_preguntas) or die('Error en la consulta
SQL'. pg_last_error($conexion))
```

Definición en la línea 119 del archivo indexEnc.php.

```
$query_cod_ciiu = "SELECT DISTINCT * FROM ciiu ORDER BY cod_ciiu ASC"
```

Definición en la línea 185 del archivo indexEnc.php.

```
$query_cod_cpc = "SELECT DISTINCT * FROM cpc ORDER BY cod_cpc ASC"
```

Definición en la línea 206 del archivo indexEnc.php.

```
if ($totalRows_preguntas > 0) if ($totalRows_secciones > 0) if
(isset($_REQUEST["id"])) $query_encuesta = "SELECT * FROM encuesta_enc
WHERE id_usuario = ".$row_usuario['id_usuario']." AND seccion < 16 order by
id_encuesta_enc ASC LIMIT 1"
```

```
if(isset($_REQUEST["id"])){ consultas de encuestas por código $query_cod_encuesta =
"SELECT * FROM encuesta WHERE cod_encuesta = 52"; $cod_encuesta =
pg_query($conexion, $query_cod_encuesta) or die('Error en la consulta SQL. Código de
encuesta'. pg_last_error($conexion)); $row_cod_encuesta =
pg_fetch_assoc($cod_encuesta); $totalRows_cod_encuesta =
pg_num_rows($cod_encuesta); echo $row_cod_encuesta[0]["cod_encuesta"]; }
```

Definición en la línea 158 del archivo indexEnc.php.

**\$query\_preguntas = "SELECT \* FROM pregunta"**

Definición en la línea 118 del archivo indexEnc.php.

**if (\$totalRows\_cod\_cpc > 0) \$query\_regional = "SELECT DISTINCT \* FROM regional WHERE estado = 'A' ORDER BY nom\_regional ASC"**

```
$whereCpc = ""; if(isset($_REQUEST["codCpc"])){ $whereCiiu = " WHERE cod_cpc =  
"$_REQUEST["codCpc"].""; } if(isset($_REQUEST["id"])) &&  
!isset($_REQUEST["codCpc"]){ $whereCiiu = " WHERE cod_cpc =  
"$_row_encuesta['ii_cpc_8_a'].""; } if(isset($_REQUEST["cod_actividad_economica"])) {  
$query_act_economica = "SELECT actividad_eco FROM ciiu WHERE cod_ciiu =  
"$_REQUEST["cod_actividad_economica"].""; $act_economica = pg_query($conexion,  
$query_act_economica) or die('Error en la consulta SQL'. pg_last_error($conexion));  
$reg_act_economica = pg_fetch_assoc($act_economica); $total_Reg_act_economica =  
pg_num_rows($act_economica); }
```

Definición en la línea 239 del archivo indexEnc.php.

**\$query\_secciones = "SELECT \* FROM seccion"**

Definición en la línea 112 del archivo indexEnc.php.

**if (isset(\$\_SESSION['MM\_Username'])) \$query\_usuario = sprintf("SELECT \* FROM usuario WHERE cedula = %s ORDER BY apellidos ASC", GetSQLValueString(\$colname\_usuario, "text"))**

Definición en la línea 103 del archivo indexEnc.php.

**\$query\_usuario\_enc = "SELECT \* FROM usuario WHERE tipo = 'Encuestador' AND estado = 'A' ORDER BY apellidos ASC"**

Definición en la línea 171 del archivo indexEnc.php.

**\$reg\_encuesta = pg\_fetch\_assoc(\$encuesta\_i)**

Definición en la línea 160 del archivo indexEnc.php.

**\$regional = pg\_query(\$conexion, \$query\_regional) or die('Error en la consulta SQL'. pg\_last\_error(\$conexion))**

Definición en la línea 240 del archivo indexEnc.php.

**\$row\_cod\_ciiu = pg\_fetch\_assoc(\$cod\_ciiu)**

Definición en la línea 187 del archivo indexEnc.php.

**\$row\_cod\_cpc = pg\_fetch\_assoc(\$cod\_cpc)**

Definición en la línea 208 del archivo indexEnc.php.

**\$row\_preguntas = pg\_fetch\_assoc(\$preguntas)**

Definición en la línea 120 del archivo indexEnc.php.

**\$row\_regional = pg\_fetch\_assoc(\$regional)**

Definición en la línea 241 del archivo indexEnc.php.

**\$row\_secciones = pg\_fetch\_assoc(\$secciones)**

Definición en la línea 114 del archivo indexEnc.php.

**\$row\_usuario = pg\_fetch\_assoc(\$usuario)**

Definición en la línea 105 del archivo indexEnc.php.

**\$row\_usuario\_enc = pg\_fetch\_assoc(\$usuario\_enc)**

Definición en la línea 173 del archivo indexEnc.php.

**\$secciones = pg\_query(\$conexion, \$query\_secciones) or die('Error en la consulta SQL'. pg\_last\_error(\$conexion))**

Definición en la línea 113 del archivo indexEnc.php.

**\$total\_Reg\_encuesta = pg\_num\_rows(\$encuesta\_i)**

Definición en la línea 161 del archivo indexEnc.php.

```
if (isset($_REQUEST["codProvincia"])) if
( (isset($_REQUEST["id"])&&isset($_REQUEST["codRegion"])&&isset($_REQUEST["codProvincia"])) $totalRows_canton = 0
```

Definición en la línea 280 del archivo indexEnc.php.

**\$totalRows\_cod\_ciiu = 0**

Definición en la línea 184 del archivo indexEnc.php.

**\$totalRows\_cod\_cpc = 0**

```
$whereCiiu = ""; if(isset($_REQUEST["codCiiu"])){ $whereCiiu = " WHERE cod_ciiu =
".$_REQUEST['codCiiu'].""; } if(isset($_REQUEST["id"]) &&
!isset($_REQUEST["codCiiu"])){ $whereCiiu = " WHERE cod_ciiu =
".$_row_encuesta['cod_actividad_economica'].""; } $totalRows_cod_cpc = 0; if($whereCiiu
!= ""){
```

Definición en la línea 205 del archivo indexEnc.php.

```
if (isset($_REQUEST["codCanton"])) if
( (isset($_REQUEST["id"])&&isset($_REQUEST["codRegion"])&&isset($_REQUEST["codProvincia"])&&isset($_REQUEST["codCanton"])) $totalRows_parroquia = 0
```

Definición en la línea 303 del archivo indexEnc.php.

**\$totalRows\_preguntas = pg\_num\_rows(\$preguntas)**

Definición en la línea 121 del archivo indexEnc.php.

```
if (isset($_REQUEST["codRegion"])) if
(isset($_REQUEST["id"])&&isset($_REQUEST["codRegion"]))
$totalRows_provincia = 0
```

Definición en la línea 257 del archivo indexEnc.php.

```
$totalRows_regional = pg_num_rows($regional)
```

Definición en la línea 242 del archivo indexEnc.php.

```
$totalRows_secciones = pg_num_rows($secciones)
```

Definición en la línea 115 del archivo indexEnc.php.

```
$totalRows_usuario = pg_num_rows($usuario)
```

Definición en la línea 106 del archivo indexEnc.php.

```
$totalRows_usuario_enc = 0
```

Definición en la línea 170 del archivo indexEnc.php.

```
$usuario = pg_query($conexion, $query_usuario) or die('Error en la consulta SQL'.
pg_last_error($conexion))
```

Definición en la línea 104 del archivo indexEnc.php.

```
$usuario_enc = pg_query($conexion, $query_usuario_enc) or die('Error en la
consulta SQL'. pg_last_error($conexion))
```

Definición en la línea 172 del archivo indexEnc.php.

```
$val_carga_fin = $row_usuario["carga_final"]
```

Definición en la línea 109 del archivo indexEnc.php.

```
$val_carga_ini = $row_usuario["carga_inicial"]
```

Definición en la línea 108 del archivo indexEnc.php.

```
if ($whereProvincia!="") if ($totalRows_canton > 0) $whereCanton = ""
```

Definición en la línea 296 del archivo indexEnc.php.

```
if ($whereRegion!="") if ($totalRows_provincia > 0) $whereProvincia = ""
```

Definición en la línea 273 del archivo indexEnc.php.

```
if ($totalRows_regional > 0) $whereRegion = ""
```

Definición en la línea 250 del archivo indexEnc.php.

**else**

```
Valor inicial:{
    $bandRegEncuesta=0
```

Definición en la línea 164 del archivo indexEnc.php.

```
if ($whereCanton!="") if ($totalRows_parroquia > 0)
if(isset($_REQUEST["buscar_cod_encuesta"])&&$_REQUEST["buscar_cod_encues
ta"]!="")
    if(!isset($_REQUEST["buscar_cod_encuesta"]) &&
    !$_REQUEST["buscar_cod_encuesta"] != "") {
    $singCodEncuesta=$_REQUEST['buscar_cod_encuesta'];
    $query_cod_encuesta=sprintf("SELECT * FROM encuesta WHERE
    cod_encuesta=%s",GetSQLValueString($singCodEncuesta, "text"));
    $cod_encuesta = pg_query($conexion, $query_cod_encuesta) or die('No se ha podido
    establecer la conexión'); $row_cod_encuesta = pg_fetch_assoc($cod_encuesta);
    $totalRows_cod_encuesta = pg_num_rows($cod_encuesta);
    }
```

Definición en la línea 319 del archivo indexEnc.php.

**if(\$row\_encuesta["seccion"]!= "")**

Definición en la línea 534 del archivo indexEnc.php.

**if (isset(\$\_REQUEST["id"])) if(isset(\$\_REQUEST["id"]))**

Definición en la línea 620 del archivo indexEnc.php.

```
if ($totalRows_secciones > 0) if
((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15"))
if((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15"))
```

Definición en la línea 3523 del archivo indexEnc.php.

## Referencia del Archivo: proyactinac/indexEnc2.php

### FUNCIONES

- `if((isset($_SERVER['QUERY_STRING'])&&($_SERVER['QUERY_STRING']!=""))`
- `if((isset($_GET['doLogout'])&&($_GET['doLogout']=="true")) isAuthorized ($strUsers, $strGroups, $UserName, $UserGroup)`

### Documentación de las funciones

```
if ((isset($_SERVER['QUERY_STRING'])&&($_SERVER['QUERY_STRING']!="))) if
((isset($_GET['doLogout'])&&($_GET['doLogout']=="true")) isAuthorized (
$strUsers, $strGroups, $UserName, $UserGroup)
```

Definición en la línea 28 del archivo index\_respaldo.php.

## VARIABLES

- `$cod_ciiu = pg_query($conexion, $query_cod_ciiu) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`
- `$cod_cpc = pg_query($conexion, $query_cod_cpc) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`
- `if(!((isset($_SESSION['MM_Username']))&&(isAuthorized("",  
$MM_authorizedUsers, $_SESSION['MM_Username'],  
$_SESSION['MM_UserGroup']))))`
- `if(!function_exists("GetSQLValueString")) $colname_usuario = "-1"`
- `if(isset($_REQUEST["id"])) if(isset($_REQUEST["id"])&&(($row_encuesta["seccion"] >  
=16 &&$row_encuesta["estado"]==  
'C')||(($row_usuario["id_usuario"]!=$row_encuesta["id_usuario"])&&(($row_encuesta["cod_encuest  
a"]  
< $row_usuario["carga_inicial"])||($row_encuesta["cod_encuesta"] >  
$row_usuario["carga_final"])))) $cont = 2`
- `$encuesta_i = pg_query($conexion, $query_encuesta) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`
- `$logoutAction = $_SERVER['PHP_SELF']."?doLogout=true"`
- `if(!isset($_SESSION)) $MM_authorizedUsers = ""`
- `$MM_donotCheckaccess = "true"`
- `$MM_restrictGoTo = "../index.php"`
- `$preguntas = pg_query($conexion, $query_preguntas) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`
- `$query_cod_ciiu = "SELECT DISTINCT * FROM ciiu ORDER BY cod_ciiu ASC"`
- `$query_cod_cpc = "SELECT DISTINCT * FROM cpc ORDER BY cod_cpc ASC"`
- `if($totalRows_preguntas >  
0) if($totalRows_secciones >  
0)`
- `if($row_usuario["tipo"]=="Analista"||$row_usuario["tipo"]=="Administrador"||$row_usuario["tipo"]  
=="Super  
Administrador"||$row_usuario["tipo"]=="Analista  
2") if($row_usuario["tipo"]=="Encuestador")`
- `if(isset($_REQUEST["id"])) $query_encuesta = "SELECT * FROM encuesta WHERE id_usuario =  
".$row_usuario["id_usuario"]." AND estado = 'T' order by id_encuesta ASC LIMIT 1"`
- `$query_preguntas = "SELECT * FROM pregunta"`
- `if($totalRows_cod_cpc > 0) $query_regional = "SELECT DISTINCT * FROM regional WHERE  
estado = 'A' ORDER BY nom_regional ASC"`
- `$query_secciones = "SELECT * FROM seccion"`
- `if(isset($_SESSION['MM_Username'])) $query_usuario = sprintf("SELECT * FROM usuario  
WHERE cedula = %s ORDER BY apellidos ASC", GetSQLValueString($colname_usuario, "text"))`
- `$query_usuario_enc = "SELECT * FROM usuario WHERE tipo = 'Encuestador' AND estado = 'A'  
ORDER BY apellidos ASC"`
- `$reg_encuesta = pg_fetch_assoc($encuesta_i)`
- `$regional = pg_query($conexion, $query_regional) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`
- `$row_cod_ciiu = pg_fetch_assoc($cod_ciiu)`
- `$row_cod_cpc = pg_fetch_assoc($cod_cpc)`
- `$row_preguntas = pg_fetch_assoc($preguntas)`
- `$row_regional = pg_fetch_assoc($regional)`
- `$row_secciones = pg_fetch_assoc($secciones)`
- `$row_usuario = pg_fetch_assoc($usuario)`
- `$row_usuario_enc = pg_fetch_assoc($usuario_enc)`

- `$secciones = pg_query($conexion, $query_secciones) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`
- `$total_Reg_encuesta = pg_num_rows($encuesta_i)`
- `if(isset($_REQUEST["codProvincia"]))`
- `if(isset($_REQUEST["id"])&&!isset($_REQUEST["codRegion"])&&!isset($_REQUEST["codProvincia"])) $totalRows_canton = 0`
- `if($totalRows_usuario_enc > 0) $totalRows_cod_ciiu = 0`
- `if($totalRows_cod_ciiu > 0) $totalRows_cod_cpc = 0`
- `if(isset($_REQUEST["codCanton"]))`
- `if(isset($_REQUEST["id"])&&!isset($_REQUEST["codRegion"])&&!isset($_REQUEST["codProvincia"])&&!isset($_REQUEST["codCanton"])) $totalRows_parroquia = 0`
- `$totalRows_preguntas = pg_num_rows($preguntas)`
- `if(isset($_REQUEST["codRegion"]))`
- `if(isset($_REQUEST["id"])&&!isset($_REQUEST["codRegion"])) $totalRows_provincia = 0`
- `$totalRows_regional = pg_num_rows($regional)`
- `$totalRows_secciones = pg_num_rows($secciones)`
- `$totalRows_usuario = pg_num_rows($usuario)`
- `$totalRows_usuario_enc = 0`
- `$usuario = pg_query($conexion, $query_usuario) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`
- `$usuario_enc = pg_query($conexion, $query_usuario_enc) or die('Error en la consulta SQL'.  
pg_last_error($conexion))`
- `$val_carga_fin = $row_usuario["carga_final"]`
- `$val_carga_ini = $row_usuario["carga_inicial"]`
- `if($whereProvincia!="") if($totalRows_canton > 0) $whereCanton = ""`
- `if($whereRegion!="") if($totalRows_provincia > 0) $whereProvincia = ""`
- `if($totalRows_regional > 0) $whereRegion = ""`
- `if($total_Reg_encuesta==0) else`
- `if($whereCanton!="") if($totalRows_parroquia > 0) if  
(isset($_REQUEST["buscar_cod_encuesta"])&&$_REQUEST["buscar_cod_encuesta"]!="")`
- `if (isset($_REQUEST["id"])&&$row_encuesta["seccion"]!="")`
- `if(isset($_REQUEST["id"])) if (isset($_REQUEST["id"]))`
- `if($totalRows_secciones >`
- `0) if((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15")) if  
((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15"))`

### ***Documentación de las variables***

**`$cod_ciiu = pg_query($conexion, $query_cod_ciiu) or die('Error en la consulta SQL'. pg_last_error($conexion))`**

Definición en la línea 195 del archivo indexEnc2.php.

**`$cod_cpc = pg_query($conexion, $query_cod_cpc) or die('Error en la consulta SQL'. pg_last_error($conexion))`**

Definición en la línea 216 del archivo indexEnc2.php.

```

if ((!(isset($_SESSION['MM_Username']))&&(isAuthorized("",
$MM_authorizedUsers, $_SESSION['MM_Username'],
$_SESSION['MM_UserGroup'])))) if (!function_exists("GetSQLValueString"))
$colname_usuario = "-1"

```

Definición en la línea 98 del archivo indexEnc2.php.

```

if (isset($_REQUEST["id"])) if
(isset($_REQUEST["id"])&&(($row_encuesta["seccion"]
>=16
&&$row_encuesta["estado"]==
'C')| |(($row_usuario["id_usuario"]!=$row_encuesta["id_usuario"])&&(($row_enc
uesta["cod_encuesta"]<
$row_usuario["carga_inicial"]| |($row_encuesta["cod_encuesta"]
>
$row_usuario["carga_final"])))))) $cont = 2

```

Definición en la línea 643 del archivo indexEnc2.php.

```

$encuesta_i = pg_query($conexion, $query_encuesta) or die('Error en la consulta
SQL'. pg_last_error($conexion))

```

Definición en la línea 168 del archivo indexEnc2.php.

```

$logoutAction = $_SERVER['PHP_SELF']."?doLogout=true"

```

Definición en la línea 10 del archivo indexEnc2.php.

```

if (!isset($_SESSION)) $MM_authorizedUsers = ""

```

Definición en la línea 6 del archivo indexEnc2.php.

```

$MM_donotCheckaccess = "true"

```

Definición en la línea 7 del archivo indexEnc2.php.

```

$MM_restrictGoTo = "../index.php"

```

Definición en la línea 53 del archivo indexEnc2.php.

```

$preguntas = pg_query($conexion, $query_preguntas) or die('Error en la consulta
SQL'. pg_last_error($conexion))

```

Definición en la línea 119 del archivo indexEnc2.php.

```

$query_cod_ciiu = "SELECT DISTINCT * FROM ciiu ORDER BY cod_ciiu ASC"

```

Definición en la línea 194 del archivo indexEnc2.php.

```
$query_cod_cpc = "SELECT DISTINCT * FROM cpc ORDER BY cod_cpc ASC"
```

Definición en la línea 215 del archivo indexEnc2.php.

```
if ($totalRows_preguntas > 0) if ($totalRows_secciones > 0) if ($row_usuario["tipo"]=="Analista" || $row_usuario["tipo"]=="Administrador" || $row_usuario["tipo"]=="Super Administrador" || $row_usuario["tipo"]=="Analista 2") if ($row_usuario["tipo"]=="Encuestador") if (isset($_REQUEST["id"])) $query_encuesta = "SELECT * FROM encuesta WHERE id_usuario = ".$row_usuario['id_usuario']."' AND estado = 'I' order by id_encuesta ASC LIMIT 1"
```

```
if(isset($_REQUEST["id"])){ consultas de encuestas por código $query_cod_encuesta = "SELECT * FROM encuesta WHERE cod_encuesta = 52"; $cod_encuesta = pg_query($conexion, $query_cod_encuesta) or die('Error en la consulta SQL. Código de encuesta'. pg_last_error($conexion)); $row_cod_encuesta = pg_fetch_assoc($cod_encuesta); $totalRows_cod_encuesta = pg_num_rows($cod_encuesta); echo $row_cod_encuesta[0]["cod_encuesta"]; }
```

Definición en la línea 167 del archivo indexEnc2.php.

```
$query_preguntas = "SELECT * FROM pregunta"
```

Definición en la línea 118 del archivo indexEnc2.php.

```
if ($totalRows_cod_cpc > 0) $query_regional = "SELECT DISTINCT * FROM regional WHERE estado = 'A' ORDER BY nom_regional ASC"
```

```
$whereCpc = ""; if(isset($_REQUEST["codCpc"])){ $whereCiiu = " WHERE cod_cpc = ".$_REQUEST['codCpc'].""; } if(isset($_REQUEST["id"]) && !isset($_REQUEST["codCpc"])){ $whereCiiu = " WHERE cod_cpc = ".$_row_encuesta['ii_cpc_8_a'].""; } if(isset($_REQUEST["cod_actividad_economica"])) { $query_act_economica = "SELECT actividad_eco FROM ciiu WHERE cod_ciiu = ".$_REQUEST["cod_actividad_economica"].""; $act_economica = pg_query($conexion, $query_act_economica) or die('Error en la consulta SQL'. pg_last_error($conexion)); $reg_act_economica = pg_fetch_assoc($act_economica); $total_Reg_act_economica = pg_num_rows($act_economica); }
```

Definición en la línea 248 del archivo indexEnc2.php.

```
$query_secciones = "SELECT * FROM seccion"
```

Definición en la línea 112 del archivo indexEnc2.php.

```
if (isset($_SESSION['MM_Username'])) $query_usuario = sprintf("SELECT * FROM usuario WHERE cedula = %s ORDER BY apellidos ASC", GetSQLValueString($colname_usuario, "text"))
```

Definición en la línea 103 del archivo indexEnc2.php.

```
$query_usuario_enc = "SELECT * FROM usuario WHERE tipo = 'Encuestador' AND estado = 'A' ORDER BY apellidos ASC"
```

Definición en la línea 180 del archivo indexEnc2.php.

```
$reg_encuesta = pg_fetch_assoc($encuesta_i)
```

Definición en la línea 169 del archivo indexEnc2.php.

```
$regional = pg_query($conexion, $query_regional) or die('Error en la consulta SQL'.  
pg_last_error($conexion))
```

Definición en la línea 249 del archivo indexEnc2.php.

```
$row_cod_ciu = pg_fetch_assoc($cod_ciu)
```

Definición en la línea 196 del archivo indexEnc2.php.

```
$row_cod_cpc = pg_fetch_assoc($cod_cpc)
```

Definición en la línea 217 del archivo indexEnc2.php.

```
$row_preguntas = pg_fetch_assoc($preguntas)
```

Definición en la línea 120 del archivo indexEnc2.php.

```
$row_regional = pg_fetch_assoc($regional)
```

Definición en la línea 250 del archivo indexEnc2.php.

```
$row_secciones = pg_fetch_assoc($secciones)
```

Definición en la línea 114 del archivo indexEnc2.php.

```
$row_usuario = pg_fetch_assoc($usuario)
```

Definición en la línea 105 del archivo indexEnc2.php.

```
$row_usuario_enc = pg_fetch_assoc($usuario_enc)
```

Definición en la línea 182 del archivo indexEnc2.php.

```
$secciones = pg_query($conexion, $query_secciones) or die('Error en la consulta  
SQL'. pg_last_error($conexion))
```

Definición en la línea 113 del archivo indexEnc2.php.

```
$total_Reg_encuesta = pg_num_rows($encuesta_i)
```

Definición en la línea 170 del archivo indexEnc2.php.

```
if (isset($_REQUEST["codProvincia"])) if
(
isset($_REQUEST["id"])&&isset($_REQUEST["codRegion"])&&isset($_REQUEST['
codProvincia'])) $totalRows_canton = 0
```

Definición en la línea 289 del archivo indexEnc2.php.

```
$totalRows_cod_ciiu = 0
```

Definición en la línea 193 del archivo indexEnc2.php.

```
$totalRows_cod_cpc = 0
```

```
$whereCiiu = ""; if(isset($_REQUEST["codCiiu"])){ $whereCiiu = " WHERE cod_ciiu =
".$_REQUEST['codCiiu'].""; } if(isset($_REQUEST["id"]) &&
!isset($_REQUEST["codCiiu"])){ $whereCiiu = " WHERE cod_ciiu =
".$_row_encuesta['cod_actividad_economica'].""; } $totalRows_cod_cpc = 0; if($whereCiiu
!= ""){
```

Definición en la línea 214 del archivo indexEnc2.php.

```
if (isset($_REQUEST["codCanton"])) if
(
isset($_REQUEST["id"])&&isset($_REQUEST["codRegion"])&&isset($_REQUEST['c
odProvincia'])&&isset($_REQUEST['codCanton'])) $totalRows_parroquia = 0
```

Definición en la línea 312 del archivo indexEnc2.php.

```
$totalRows_preguntas = pg_num_rows($preguntas)
```

Definición en la línea 121 del archivo indexEnc2.php.

```
if (isset($_REQUEST["codRegion"])) if
(
isset($_REQUEST["id"])&&isset($_REQUEST["codRegion"]))
$totalRows_provincia = 0
```

Definición en la línea 266 del archivo indexEnc2.php.

```
$totalRows_regional = pg_num_rows($regional)
```

Definición en la línea 251 del archivo indexEnc2.php.

```
$totalRows_secciones = pg_num_rows($secciones)
```

Definición en la línea 115 del archivo indexEnc2.php.

```
$totalRows_usuario = pg_num_rows($usuario)
```

Definición en la línea 106 del archivo indexEnc2.php.

```
$totalRows_usuario_enc = 0
```

Definición en la línea 179 del archivo indexEnc2.php.

```
$usuario = pg_query($conexion, $query_usuario) or die('Error en la consulta SQL'.  
pg_last_error($conexion))
```

Definición en la línea 104 del archivo indexEnc2.php.

```
$usuario_enc = pg_query($conexion, $query_usuario_enc) or die('Error en la  
consulta SQL'. pg_last_error($conexion))
```

Definición en la línea 181 del archivo indexEnc2.php.

```
$val_carga_fin = $row_usuario["carga_final"]
```

Definición en la línea 109 del archivo indexEnc2.php.

```
$val_carga_ini = $row_usuario["carga_inicial"]
```

Definición en la línea 108 del archivo indexEnc2.php.

```
if ($whereProvincia!="") if ($totalRows_canton > 0) $whereCanton = ""
```

Definición en la línea 305 del archivo indexEnc2.php.

```
if ($whereRegion!="") if ($totalRows_provincia > 0) $whereProvincia = ""
```

Definición en la línea 282 del archivo indexEnc2.php.

```
if ($totalRows_regional > 0) $whereRegion = ""
```

Definición en la línea 259 del archivo indexEnc2.php.

**else**

```
Valor inicial:{  
$bandRegEncuesta=0
```

Definición en la línea 173 del archivo indexEnc2.php.

```
if ($whereCanton!="") if ($totalRows_parroquia > 0)  
if(isset($_REQUEST["buscar_cod_encuesta"])&&$_REQUEST["buscar_cod_encues  
ta"]!="")
```

```
if(!isset($_REQUEST["buscar_cod_encuesta"])) &&  
!$_REQUEST["buscar_cod_encuesta"] != "" ) {  
$ingCodEncuesta=$_REQUEST['buscar_cod_encuesta'];  
$query_cod_encuesta=sprintf("SELECT * FROM encuesta WHERE  
cod_encuesta=%s",GetSQLValueString($ingCodEncuesta, "text"));  
$cod_encuesta = pg_query($conexion, $query_cod_encuesta) or die('No se ha podido  
establecer la conexión'); $row_cod_encuesta = pg_fetch_assoc($cod_encuesta);  
$totalRows_cod_encuesta = pg_num_rows($cod_encuesta);  
}
```

Definición en la línea 328 del archivo indexEnc2.php.

```
if(isset($_REQUEST["id"])&&$row_encuesta["seccion"]!= '')
```

Definición en la línea 544 del archivo indexEnc2.php.

```
if (isset($_REQUEST["id"])) if(isset($_REQUEST["id"]))
```

Definición en la línea 630 del archivo indexEnc2.php.

```
if ($totalRows_secciones > 0) if  
((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15"))  
if((isset($row_encuesta["seccion"])&&$row_encuesta["seccion"] >="15"))
```

Definición en la línea 3541 del archivo indexEnc2.php.

## Referencia del Archivo: proyactinac/indexRep.php

### FUNCIONES

- [if](#)((isset(\$\_SERVER['QUERY\_STRING'])&&(\$\_SERVER['QUERY\_STRING']!=""))) )
- [if](#)((isset(\$\_GET['doLogout'])&&(\$\_GET['doLogout']=="true"))) [isAuthorized](#) (\$strUsers, \$strGroups, \$UserName, \$UserGroup)

#### Documentación de las funciones

```
if ((isset($_SERVER['QUERY_STRING'])&&($_SERVER['QUERY_STRING']!=""))) if  
((isset($_GET['doLogout'])&&($_GET['doLogout']=="true"))) isAuthorized (  
$strUsers, $strGroups, $UserName, $UserGroup)
```

Definición en la línea 28 del archivo indexRep.php.

### VARIABLES

- [if](#)(!(isset(\$\_SESSION['MM\_Username'])&&(isAuthorized("", \$MM\_authorizedUsers, \$\_SESSION['MM\_Username'], \$\_SESSION['MM\_UserGroup']))))
- [if](#)(!function\_exists("GetSQLValueString")) [\\$colname\\_usuario](#) = "-1"
- [\\$logoutAction](#) = \$\_SERVER['PHP\_SELF']."?doLogout=true"
- [if](#)(isset(\$\_SESSION)) [\\$MM\\_authorizedUsers](#) = ""
- [\\$MM\\_donotCheckaccess](#) = "true"
- [\\$MM\\_restrictGoTo](#) = "../index.php"
- [\\$query\\_regional](#) = "SELECT DISTINCT \* FROM regional WHERE estado='A' ORDER BY nom\_regional ASC"

- `if($totalRows_regional > 0) $query_seccion = "SELECT DISTINCT * FROM seccion ORDER BY descripcion ASC"`
- `if(isset($_SESSION['MM_Username'])) $query_usuario = sprintf("SELECT * FROM usuario WHERE cedula = %s ORDER BY apellidos ASC", GetSQLValueString($colname_usuario, "text"))`
- `$regional = pg_query($conexion, $query_regional) or die('Error en la consulta SQL'. pg_last_error($conexion))`
- `$row_regional = pg_fetch_assoc($regional)`
- `$row_seccion = pg_fetch_assoc($seccion)`
- `$row_usuario = pg_fetch_assoc($usuario)`
- `$seccion = pg_query($conexion, $query_seccion) or die('Error en la consulta SQL'. pg_last_error($conexion))`
- `$totalRows_regional = pg_num_rows($regional)`
- `$totalRows_seccion = pg_num_rows($seccion)`
- `$totalRows_usuario = pg_num_rows($usuario)`
- `$usuario = pg_query($conexion, $query_usuario) or die('Error en la consulta SQL'. pg_last_error($conexion))`
- `if($totalRows_seccion > 0) if (isset($_REQUEST["cod_encuesta"])&&$_REQUEST["cod_encuesta"]!="")`

### Documentación de las variables

```

if (!((isset($_SESSION['MM_Username'])))&&(isAuthorized("",
$MM_authorizedUsers, $_SESSION['MM_Username'],
$_SESSION['MM_UserGroup'])))) if (!function_exists("GetSQLValueString"))
$colname_usuario = "-1"
    
```

Definición en la línea 98 del archivo indexRep.php.  
**\$logoutAction = \$\_SERVER['PHP\_SELF']."?doLogout=true"**

Definición en la línea 10 del archivo indexRep.php.  
**if (!isset(\$\_SESSION)) \$MM\_authorizedUsers = ""**

Definición en la línea 6 del archivo indexRep.php.  
**\$MM\_donotCheckaccess = "true"**

Definición en la línea 7 del archivo indexRep.php.  
**\$MM\_restrictGoTo = "../index.php"**

Definición en la línea 53 del archivo indexRep.php.  
**\$query\_regional = "SELECT DISTINCT \* FROM regional WHERE estado='A' ORDER BY nom\_regional ASC"**

Definición en la línea 109 del archivo indexRep.php.  
**if (\$totalRows\_regional > 0) \$query\_seccion = "SELECT DISTINCT \* FROM seccion ORDER BY descripcion ASC"**

Definición en la línea 122 del archivo indexRep.php.  
**if (isset(\$\_SESSION['MM\_Username'])) \$query\_usuario = sprintf("SELECT \* FROM usuario WHERE cedula = %s ORDER BY apellidos ASC", GetSQLValueString(\$colname\_usuario, "text"))**

Definición en la línea 103 del archivo indexRep.php.

```
$regional = pg_query($conexion, $query_regional) or die('Error en la consulta SQL'.  
pg_last_error($conexion))
```

Definición en la línea 110 del archivo indexRep.php.

```
$row_regional = pg_fetch_assoc($regional)
```

Definición en la línea 111 del archivo indexRep.php.

```
$row_seccion = pg_fetch_assoc($seccion)
```

Definición en la línea 124 del archivo indexRep.php.

```
$row_usuario = pg_fetch_assoc($usuario)
```

Definición en la línea 105 del archivo indexRep.php.

```
$seccion = pg_query($conexion, $query_seccion) or die('Error en la consulta SQL'.  
pg_last_error($conexion))
```

Definición en la línea 123 del archivo indexRep.php.

```
$totalRows_regional = pg_num_rows($regional)
```

Definición en la línea 112 del archivo indexRep.php.

```
$totalRows_seccion = pg_num_rows($seccion)
```

Definición en la línea 125 del archivo indexRep.php.

```
$totalRows_usuario = pg_num_rows($usuario)
```

Definición en la línea 106 del archivo indexRep.php.

```
$usuario = pg_query($conexion, $query_usuario) or die('Error en la consulta SQL'.  
pg_last_error($conexion))
```

Definición en la línea 104 del archivo indexRep.php.

```
if ($totalRows_seccion > 0)  
if(isset($_REQUEST["cod_encuesta"])&&$_REQUEST["cod_encuesta"]!="")
```

Definición en la línea 135 del archivo indexRep.php.

## Referencia del Archivo: proyactinac/MMHTTPDB.php

### FUNCIONES

- [create\\_error](#) (\$error\_msg= "", \$line=-1)
- [log\\_messages](#) (\$error\_message)

### ***Documentación de las funciones***

#### **create\_error ( \$error\_msg = '', \$line = -1)**

Definición en la línea 19 del archivo MMHTTPDB.php.  
Referenciado por MySqlConnection\HandleException().

#### **log\_messages ( \$error\_message)**

Definición en la línea 5 del archivo MMHTTPDB.php.  
Referenciado por MySqlConnection\HandleException().

## ***VARIABLES***

- [\\$debug\\_to\\_file](#) = false
- foreach(\$\_POST as \$key=>
- \$value) [if](#)(isset(\$\_POST['opCode'])&&\$\_POST['opCode']==
- 'IsOpen') [if](#)(extension\_loaded('mbstring')) [else](#)
- [return](#)

### ***Documentación de las variables***

#### **\$debug\_to\_file = false**

Definición en la línea 3 del archivo MMHTTPDB.php.  
Referenciado por MySqlConnection\HandleException() y log\_messages().

#### **[if](#) (isset(\$oConn)&&\$oConn) else**

```
valor inicial:{  
    echo '<html><head></head>'
```

Definición en la línea 53 del archivo MMHTTPDB.php.

#### **return**

Definición en la línea 70 del archivo MMHTTPDB.php.  
Referenciado por close().

## ESTRUCTURAS DE DATOS

- class [MySQLConnection](#)

## FUNCIONES

- [if\(!function\\_exists\('mysql\\_connect'\)||!function\\_exists\('mysql\\_pconnect'\)||!extension\\_loaded\('mysql'\)\)](#)  
[KT\\_ErrorHandler](#) (\$errno, \$errstr, \$errfile, \$errline)

### Documentación de las funciones

#### [if](#)

**(!function\_exists('mysql\_connect') || !function\_exists('mysql\_pconnect') || !extension\_loaded('mysql')) KT\_ErrorHandler ( \$errno, \$errstr, \$errfile, \$errline)**

Definición en la línea 17 del archivo mysql.php.

## VARIABLES

- const [CONN\\_NOT\\_OPEN\\_GET\\_DB\\_LIST\\_create\\_error](#)("The Connection could not be opened when trying to retrieve the database list.")
- const [CONN\\_NOT\\_OPEN\\_GET\\_TABLES\\_create\\_error](#)("The Connection could not be opened when trying to retrieve the tables.")
- const [MYSQL\\_NOT\\_EXISTS](#) (!function\_exists('create\_error')) [create\\_error](#)("Your PHP server doesn't have the MySQL module loaded or you can't use the mysql\_(p)connect functions.")

## Bibliografía

Arias, S. (15 de 11 de 2013). API-Documentacion Técnica ACTI. *index.html* . Quito, Quito, Ecuador: DASI.

---

<sup>i</sup> (Arias, 2013)