**Integrante:**

\* Martin González Hernández

**Universidad Tecnológica de la Riviera Maya**

**TI51**

**TSU-Quinto Cuatrimestre**



Proyecto: MiApp.Clientes

**Estándares de Codificación**

**2015**

Contenido

[Convención de nombres para el código 3](#_Toc411884129)

[Variables 3](#_Toc411884130)

[Constantes 3](#_Toc411884131)

[Métodos y Funciones 3](#_Toc411884132)

[Clases 3](#_Toc411884133)

[Convención de nombres para las tablas de la base de datos 4](#_Toc411884134)

[Etiquetas de php 4](#_Toc411884135)

[Inclusión de archivos 4](#_Toc411884136)

[Comentarios 4](#_Toc411884137)

[Estructuras de control 5](#_Toc411884138)

[Estructura if 5](#_Toc411884139)

[If anidado 6](#_Toc411884140)

[Dividir sentencias if largas 6](#_Toc411884141)

[Estructura While 7](#_Toc411884142)

[Operador ternario 7](#_Toc411884143)

Este documento representa la declaración del estándar de codificación que se usara para codificar mi aplicación.

# Convención de nombres para el código

## Variables

* + Los nombres de las variables se escribirán en minúsculas.
  + Los nombres de las viables que se compongan de dos o más palabras se escribirán en formato Camel Case.
  + Para los nombres de las variables no se usaran caracteres especiales.
  + Los nombres de las variables solo se escribirán con letras de la a-z.

## Constantes

* + Las constantes deben ser definidas en mayusculas y utilizando guion bajo (\_) cómo separador
  + Los nombres de las constantes solo se escribirán con letras de la A-Z.
  + Para los nombres de las constantes no se usaran caracteres especiales.

## Métodos y Funciones

* Los métodos y funciones que se compongan de dos o más palabras deben ser escritos utilizando la técnica camelCase o un guion bajo (\_) como separador.

## Clases

* Las clases se deberán escribir con una mayúscula al principio para identificarlas como principales.

# Convención de nombres para las tablas de la base de datos

* Se utilizara caracteres alfanuméricos.
* El limite los nombres a menos de 64 caracteres (limitación de MySQL).
* Se utilizara el guion bajo (\_) para separar palabras.
* Se utilizara palabras en minúsculas.
* Los nombres de las tablas deben ir en plural y los nombres de las columnas en singular.
* Se utilizara las letras id en las columnas de clave principal y externa.
* En una tabla, se utilizara en primer lugar la clave principal, seguida de las claves externas.
* Los nombres de los campos deben ser descriptivos de su contenido.
* Los nombres de los campos deben ser unívocos entre las tablas, excepción hecha en las claves.

# Etiquetas de php

Por compatibilidad solo se usara el siguiente tipo de etiquetas para indicar un bloque de PHP:

<?php ?>

# Inclusión de archivos

Para incluir archivos se usara la siguiente función.

require\_once

require

# Comentarios

Los comentarios se escribirán de la siguiente forma:

Comentarios de una sola línea: //Este es un comentario de una línea

Comentarios de más de una línea: /\* Este en un comentario

De más varias líneas \*/

# Estructuras de control

Las estructuras de control se escribirán de la siguiente manera

## 

## Estructura if

<?php

if ((condition1) || (condition2)) {

action1;

} elseif ((condition3) && (condition4)) {

action2;

} else {

defaultaction;

}

?>

<?php

If (condición) {

//acción1

}else{

//acción2

}

?>

<?php

If (condición) {

//acción

}

?>

## If anidado

<?php

If (condición) {

//acción

} if (condition2) {

//acción

}

?>

# Dividir sentencias if largas

<?php

if (($condition1

|| $condition2)

&& $condition3

&& $condition4

) {

//code here

}

?>

<?php

If (condición1) && (condición2) && (condición3) && (condicion4) && (condicion5){

//acción

}

?>

# Estructura While

While (condición) {

//acción

}

# Operador ternario

<?php

$a = $condition1 && $condition2

? $foo : $bar;

$b = $condition3 && $condition4

? $foo\_man\_this\_is\_too\_long\_what\_should\_i\_do

: $bar;

?>