

Universidad del Istmo

Facultad de Ingeniería

Manual funcional

Base de datos I

Kevin Champney

Elisa Monzon

Guatemala 18 de Octubre de 2019

Contenido

[Base de datos 1](#_Toc22287296)

[Tablas 1](#_Toc22287297)

[Cobro 1](#_Toc22287298)

[Comentarios 1](#_Toc22287299)

[Moderación 1](#_Toc22287300)

[Posts 2](#_Toc22287301)

[Subtopic 2](#_Toc22287302)

[Suscripción 2](#_Toc22287303)

[Topics 2](#_Toc22287304)

[Users 3](#_Toc22287305)

[Aplicación en páginas web - php 4](#_Toc22287306)

[Conexión 4](#_Toc22287307)

[Login 4](#_Toc22287308)

[Registro 4](#_Toc22287309)

[Cambio de password 4](#_Toc22287310)

[Administración de usuarios en general 4](#_Toc22287311)

[Manejo de perfiles registrados o anónimos 4](#_Toc22287312)

[Creación de artículos 4](#_Toc22287313)

[Categorías 4](#_Toc22287314)

[Subcategorías 4](#_Toc22287315)

[Venta suscripciones 4](#_Toc22287316)

[Funcionalidad de paga 4](#_Toc22287317)

[Búsqueda con filtros y ordenamiento 4](#_Toc22287318)

[Actualización de articulo 4](#_Toc22287319)

[Manejo de perfiles de usuario 4](#_Toc22287320)

[Funcionalidades MySQL 4](#_Toc22287321)

[Procedimiento con parámetros out 4](#_Toc22287322)

[Vistas 4](#_Toc22287323)

# Base de datos

## Tablas

### Cobro

Campos:

* Id: llave primaria de tipo int autoincremental
* Id\_users: llave foránea de tipo int, contiene la información del usuario que ha ingresado un cobro
* Id\_sus: llave foránea de tipo int, contiene la información del usuario que ha
* Tarjeta: campo de tipo int que recibe el numero de tarjeta ingresado por el usuario
* Vencimiento: campo de tipo int que recibe la fecha de vencimiento ingresada por el usuario
* Atrás: campo de tipo int que recibe los tres dígitos de la parte lateral ingresados por el usuario
* Iniciosusc: campo de tipo datetime con valor predeterminado current\_timestamp () para obtener inicio de suscripción

### Comentarios

Campos

* Id: llave principal tipo int, autoincremental
* Id\_users: llave foránea tipo int, contiene la información del usuario que ha realizado el comentario
* id\_posts: llave foránea tipo int, contiene la información del articulo en el que se ha realizado el comentario
* contenido: tipo varchar, recibe el texto ingresado por el usuario en el form de comentarios
* Created\_at: tipo datetime con valor predeterminado current\_timestamp () para obtener fecha de publicación
* vecesreporte: tipo int, lleva conteo de cuantas veces se ha hecho un reporte

### Moderación

Campos:

* id: llave principal tipo int autoincremental
* id\_comment: llave foránea tipo int, contiene la información de los comentarios, para generar reporte de moderación. Predeterminado como null, para no requerir siempre de sus valores
* id\_posts: llave foránea tipo int, contiene la información de los artículos que sean enviados a revisión. Predeterminado como null, para no requerir siempre de sus valores
* censurar: tipo tinyint, usado para marcar si un comentario o articulo han sido censurados. 0 para falso y 1 para verdadero

### Posts

Campos:

* id: llave principal de tipo int autoincremental
* user\_id: llave foránea de tipo int, contiene la información del usuario que ha creado el post
* id\_subtopic: llave foránea de tipo int, contiene la información sobre a que subcategoría pertenece
* Title: tipo varchar, recibe el título del artículo.
* Slug: tipo varchar, convierte el titulo del articulo a un texto separado por guiones, es utilizado para su búsqueda
* Views: tipo int, lleva conteo sobre cuantas veces un artículo ha sido visitado
* Image: tipo blob, recibe las imágenes que sean cargadas por el autor a su artículo, este campo lo guardará como bloc
* Body: tipo varchar, recibe la información que formará parte del cuerpo del articulo
* Published: tipo tinyint, determina si un artículo ha sido publicado o está en versión draft
* Created\_at: tipo datetime con valor predeterminado current\_timestamp () para obtener fecha de publicación
* Updated\_at: tipo datetime con valor predeterminado current\_timestamp () para obtener fecha de actualización
* Plantilla: tipo int, recibe el valor sobré qué plantilla corresponde al artículo.
* Premium: tipo tinyint, determina si un artículo es de paga o gratis.

### Subtopic

Campos:

* Id: llave principal tipo int autoincremental
* Id\_topic: llave foránea de tipo int, contiene la información sobre el topic al que ha sido ligado un subtopic
* Nombre: tipo varchar, recibe el nombre que se ha dado a la subcategoría
* Plantilla: tipo int, determina en qué plantilla se mostrará la subcategoría
* Descripción: tipo varchar, recibe una descripción sobre la subcategoría.

### Suscripción

Campos:

* Id: llave principal tipo int autoincremental
* Tipo: tipo varchar, recibe el nombre de la suscripción
* Descripción: tipo varchar, recibe una descripción sobre la suscripción
* Costo: tipo int, recibe el precio por suscripción

### Topics

Campos:

* Id: llave primaria, tipo int autoincremental
* Name: tipo varchar, recibe el nombre de la categoría
* Slug: tipo varchar, convierte el título del articulo a un texto separado por guiones, es utilizado para su búsqueda

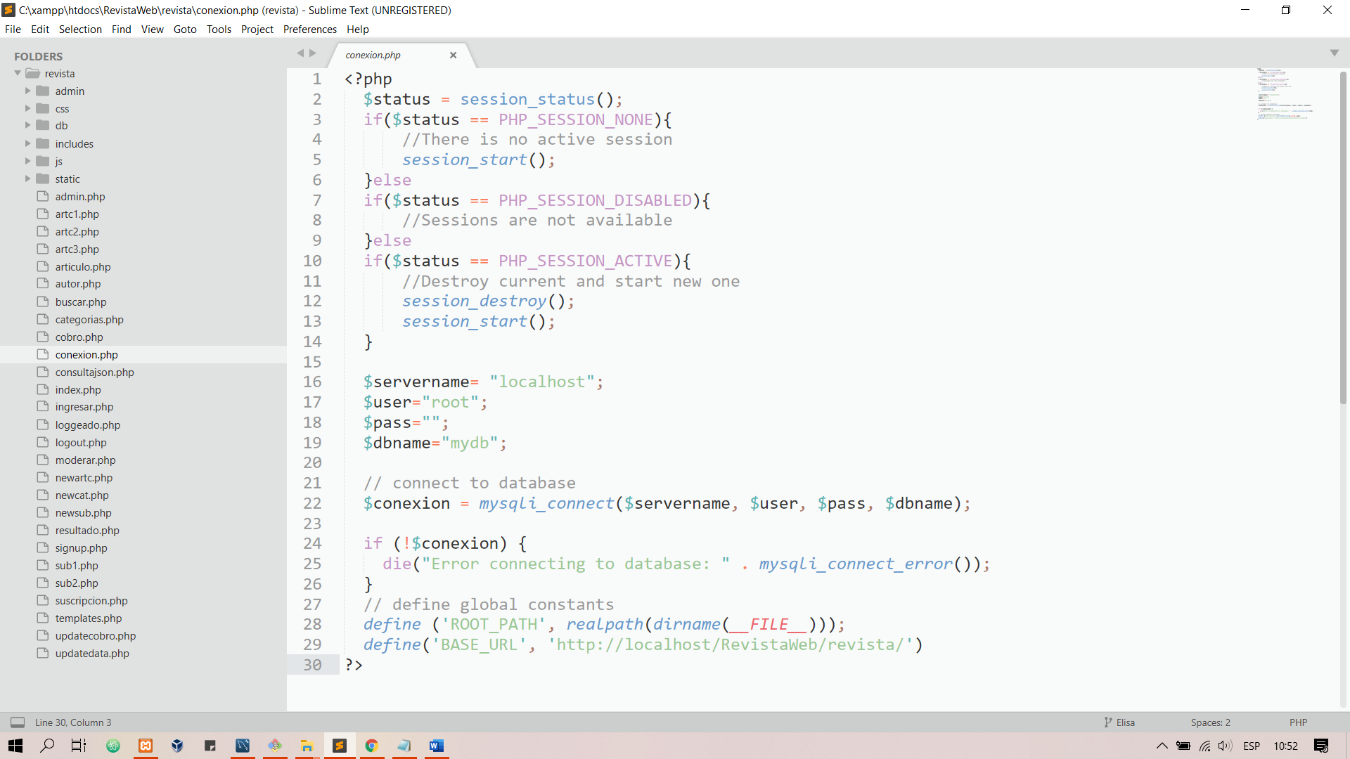
### Users

Campos:

* Id: llave primaria de tipo int autoincremental
* Nombre: tipo varchar, recibe el nombre del usuario
* Apellido: tipo varchar, recibe el apellido del usuario
* Username: tipo varchar, recibe el username del usuario
* email: tipo varchar, recibe el email del usuario
* role: tipo enum (Author, Admin, Lector, Moderador) con valor predeterminado como Null 🡪 Lector
* password: tipo varchar, recibe la contraseña codificada del usuario
* Created\_at: tipo datetime con valor predeterminado current\_timestamp () para obtener fecha de publicación
* Updated\_at: tipo datetime con valor predeterminado current\_timestamp () para obtener fecha de actualización
* Foto:
* Suscripción: tipo tinyint, determina si un usuario cuenta con suscripción.

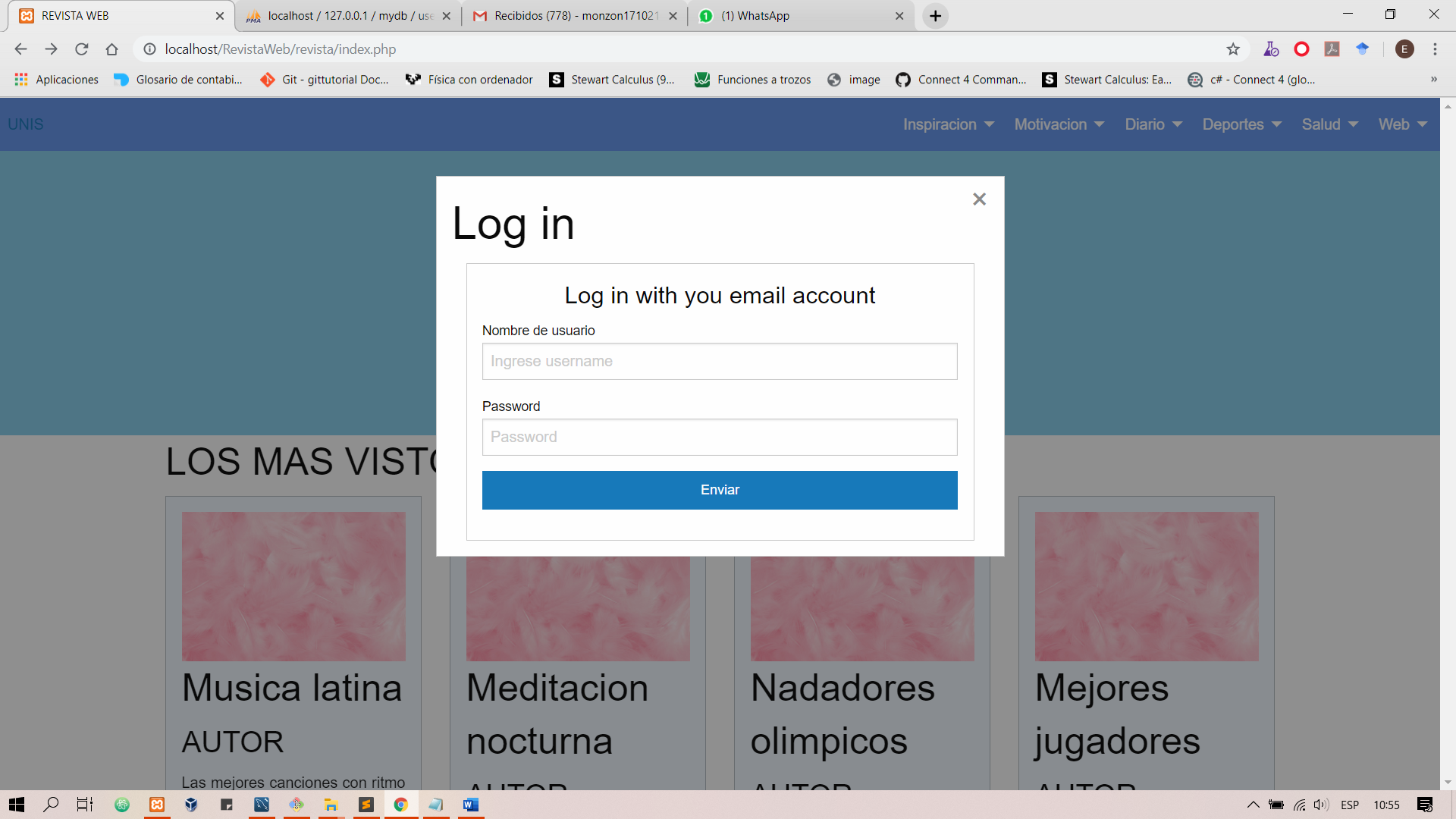
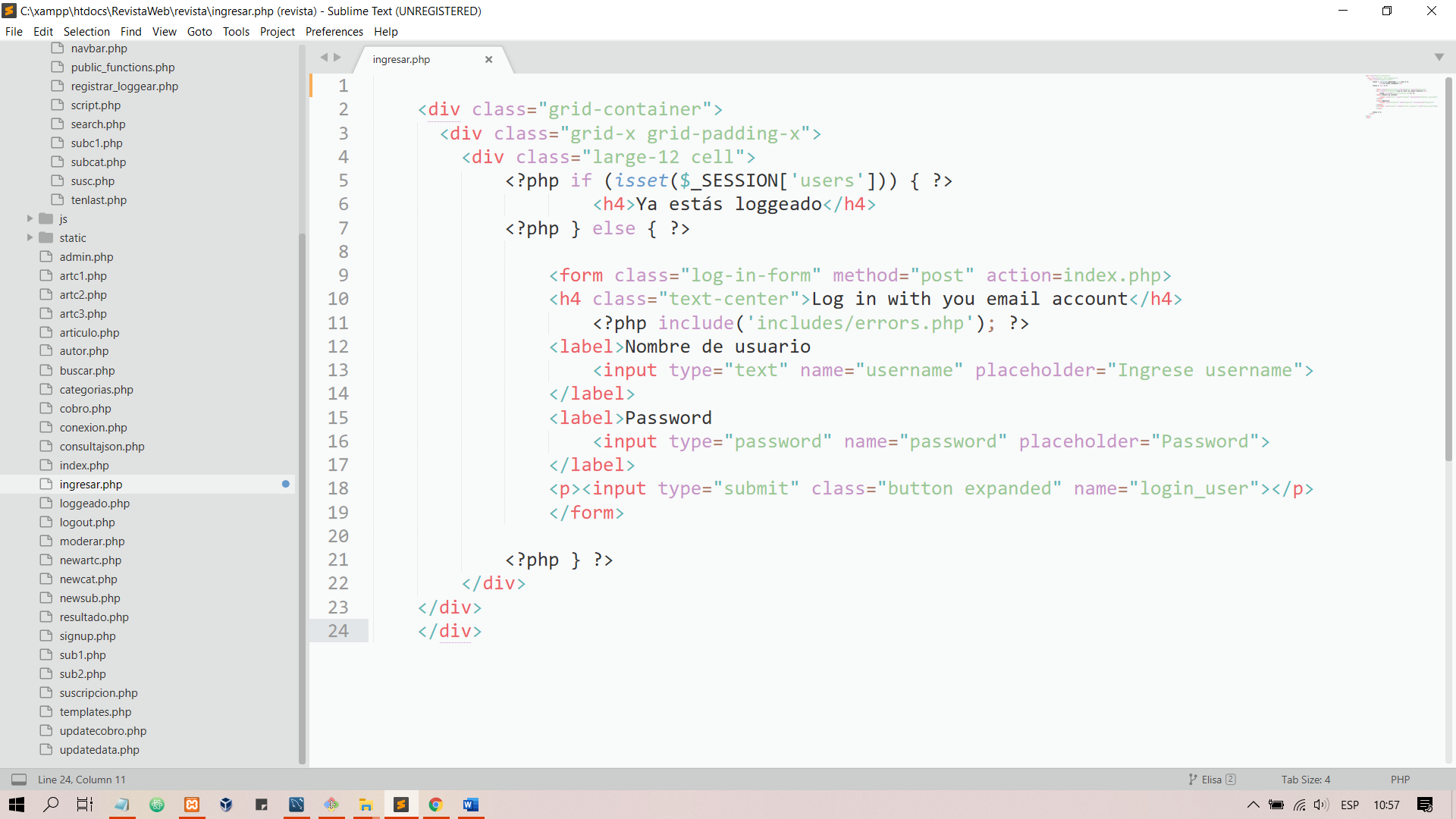
# Aplicación en páginas web - php

## Conexión

Creación de conexión con la base de datos utilizando mysqli procedural

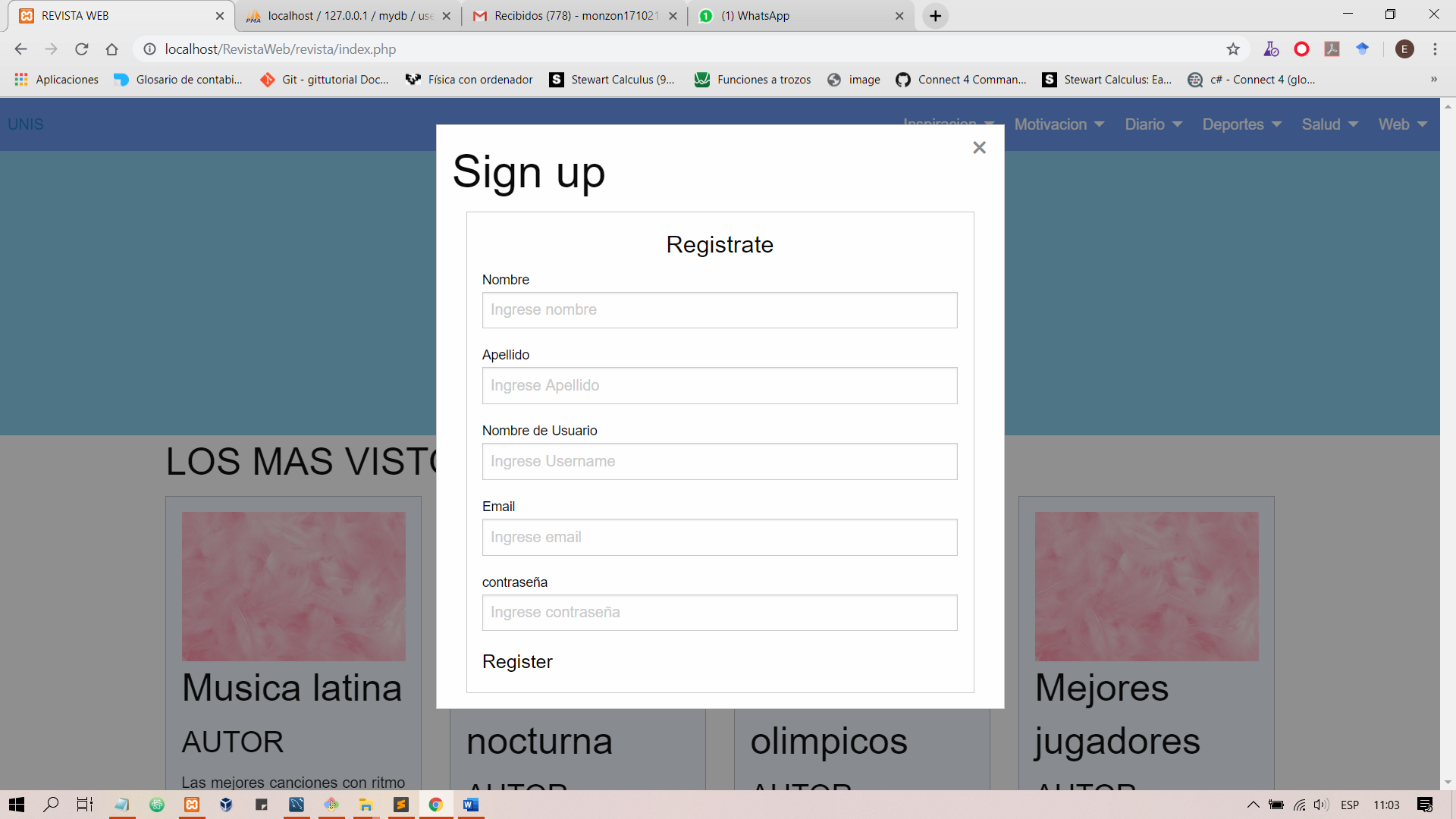
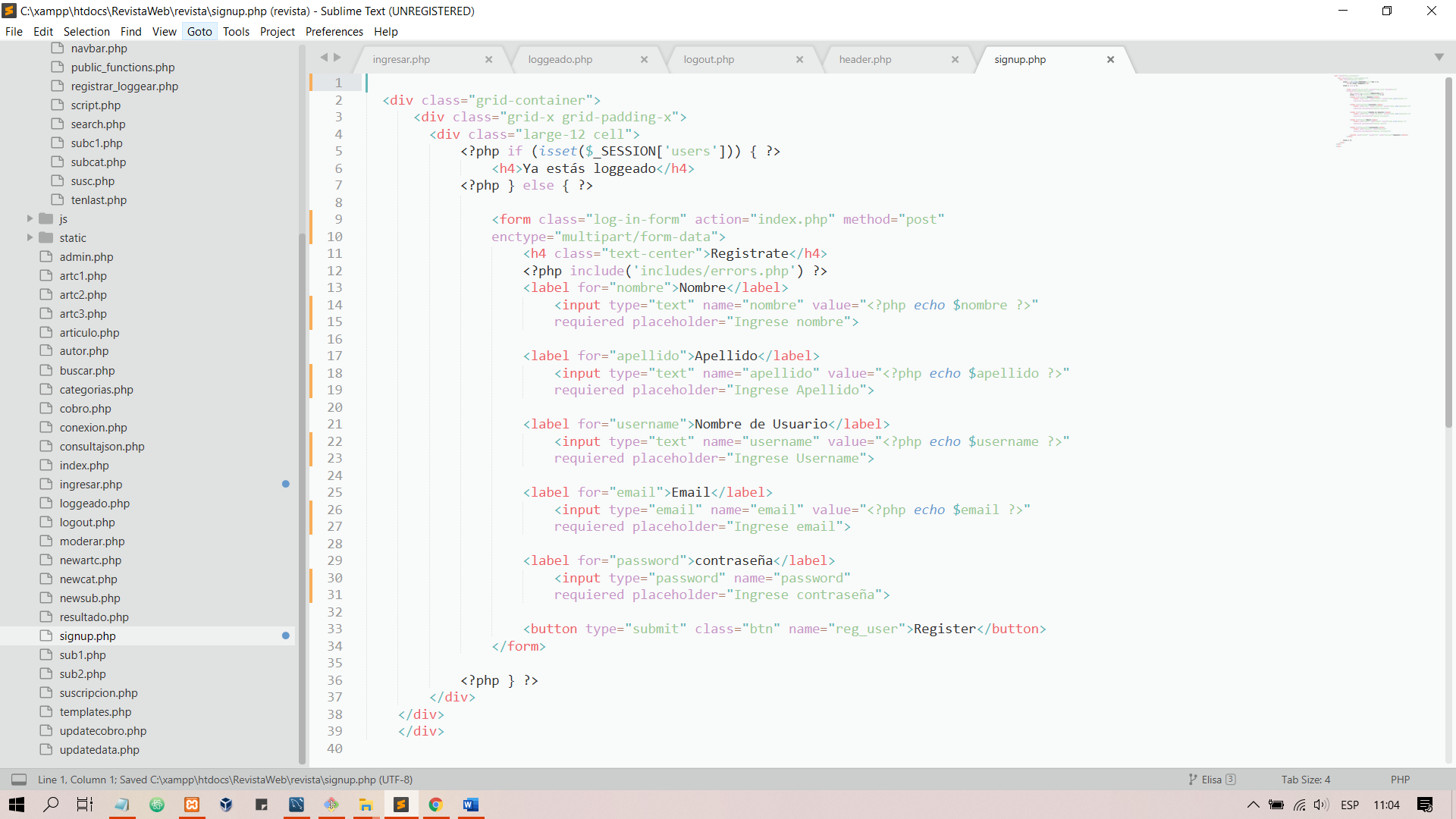
Validación de inicio y cierre de sesión

## Login

En este formulario se ingresan los datos del usuario registrado por método POST y se redirige a loggeado.php.

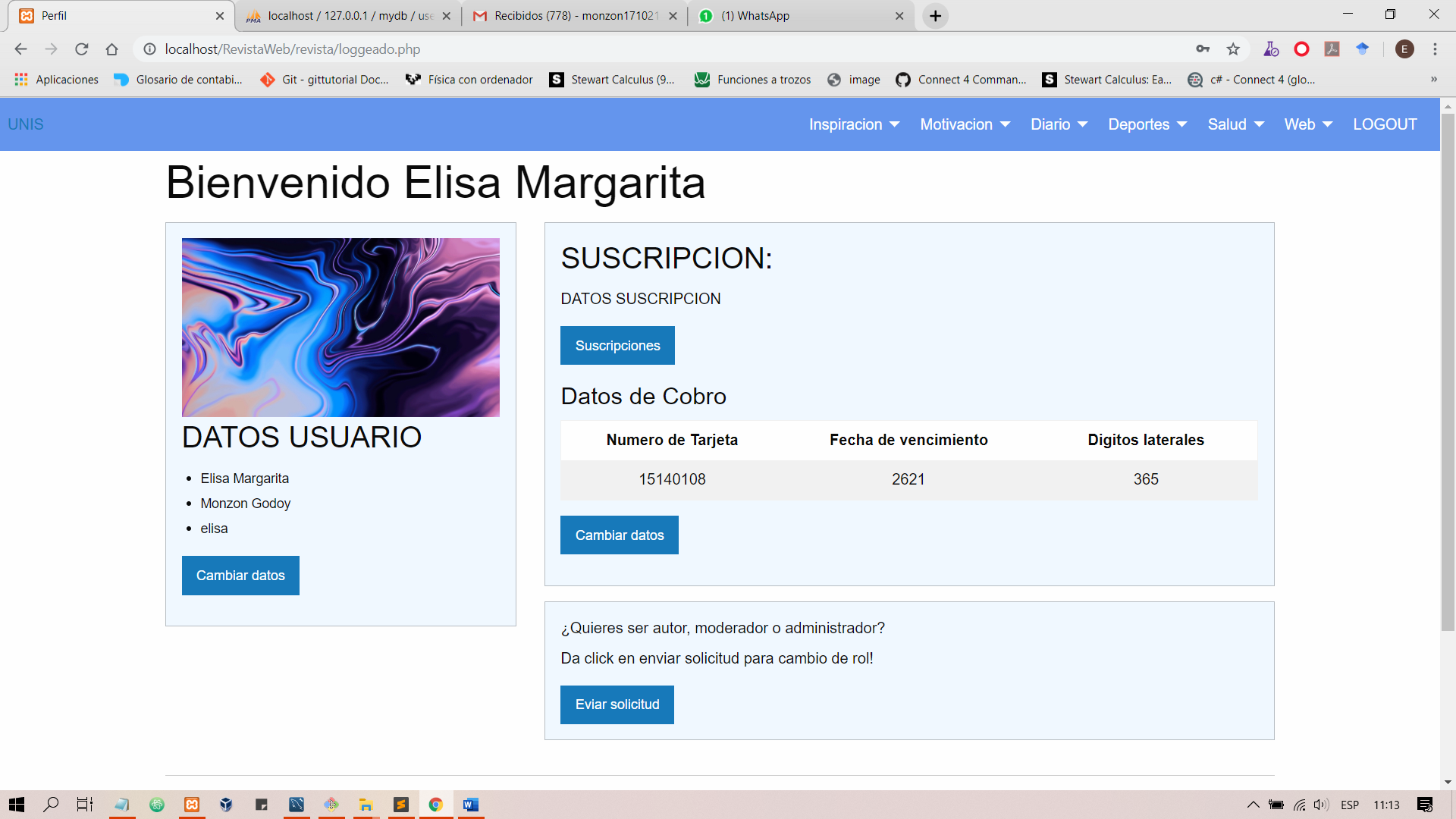
## Registro

En este formulario se ingresan los datos del usuario registrado por método POST y se redirige a loggeado.php.

## Cambio de contraseña

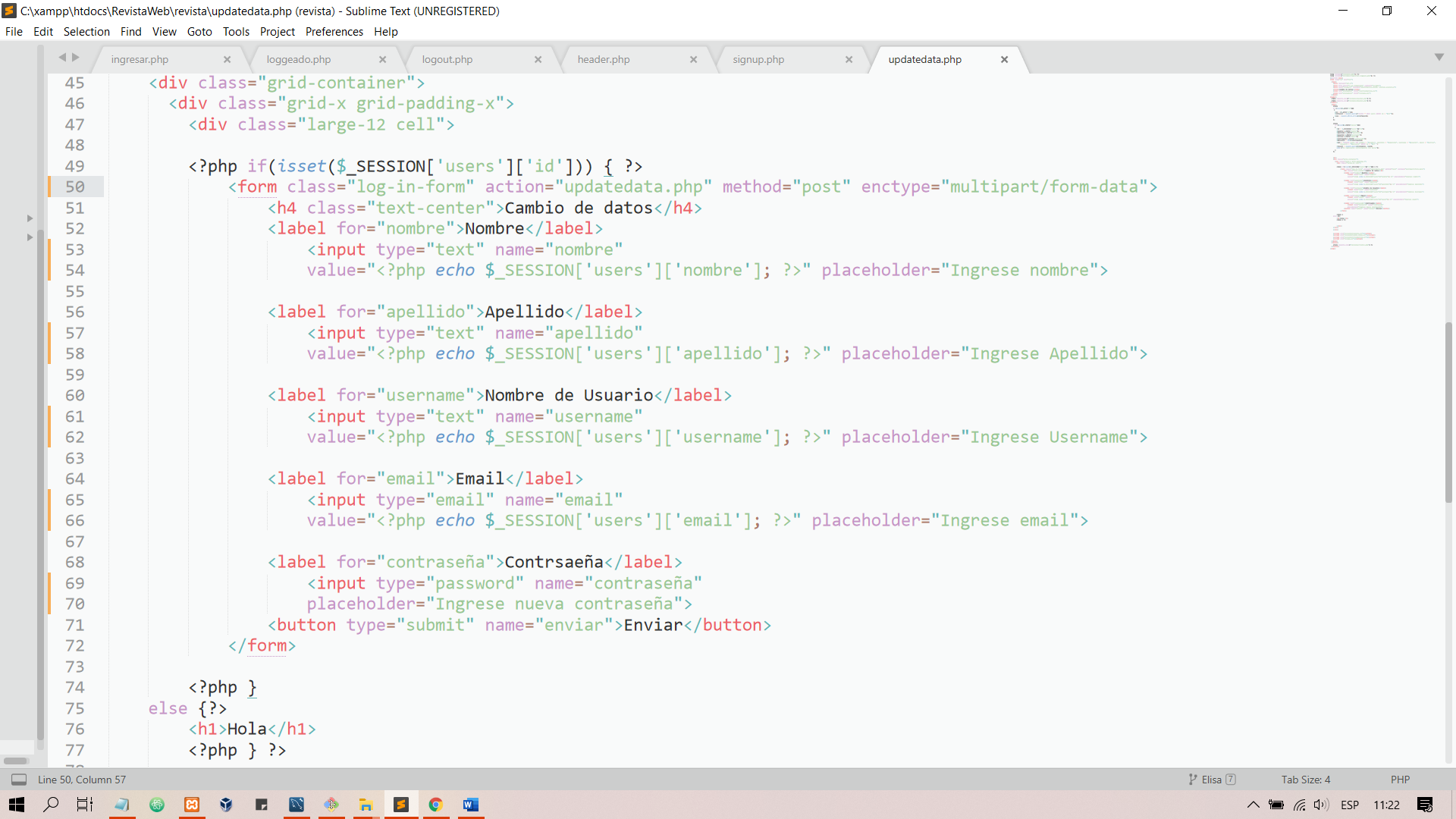
Desde el perfil se tiene un botón “Cambiar datos” con el siguiente formato:



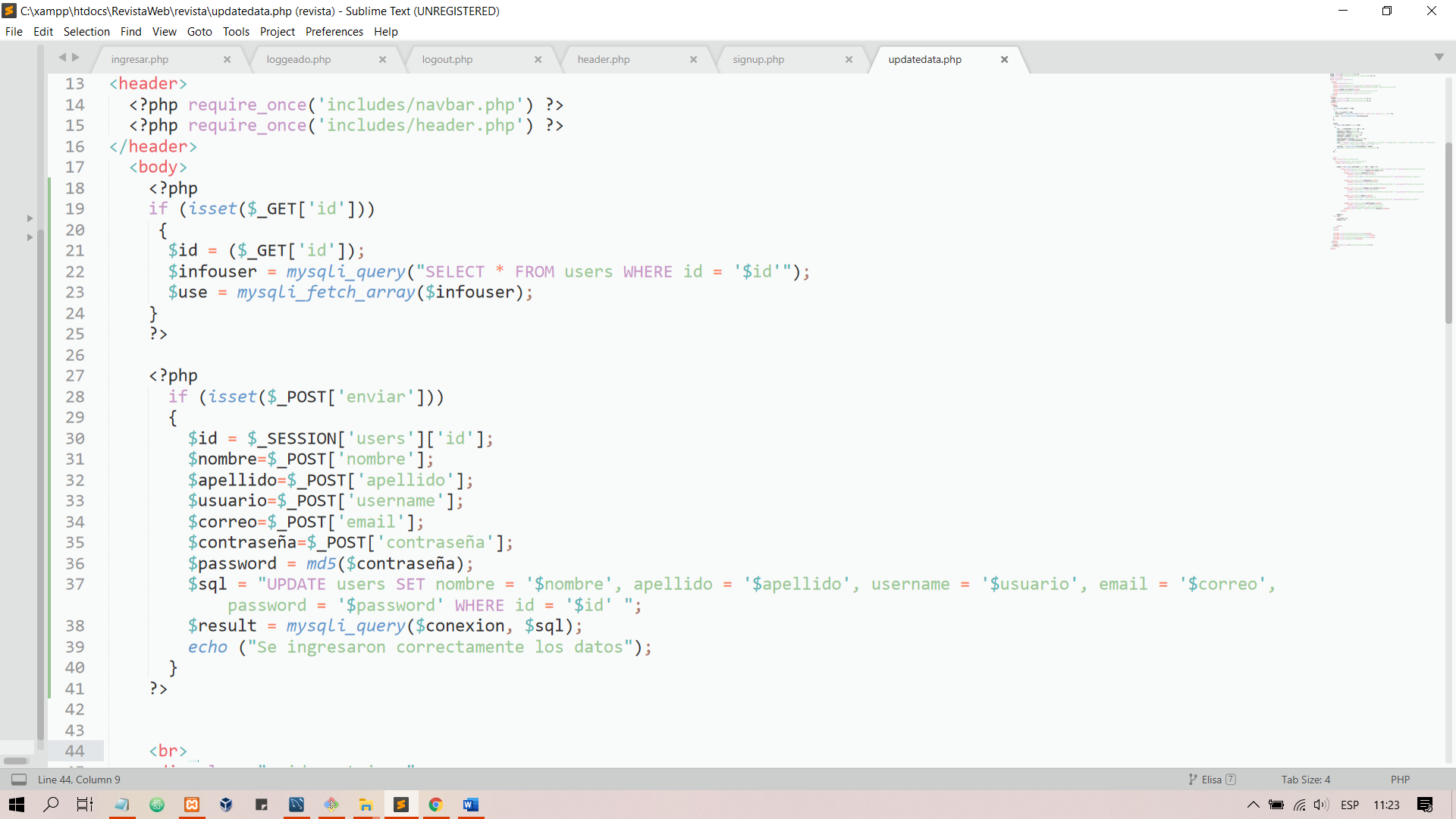
*<a href="updatedata.php?id=<?php echo $\_SESSION['users']['id'] ?>" class="button">Cambiar datos</a>*

Al dar clic en el botón se lee el id del usuario loggeado y así se puede obtener su información

El formulario para ingresar los datos es:



Las variables, código php y sql es:

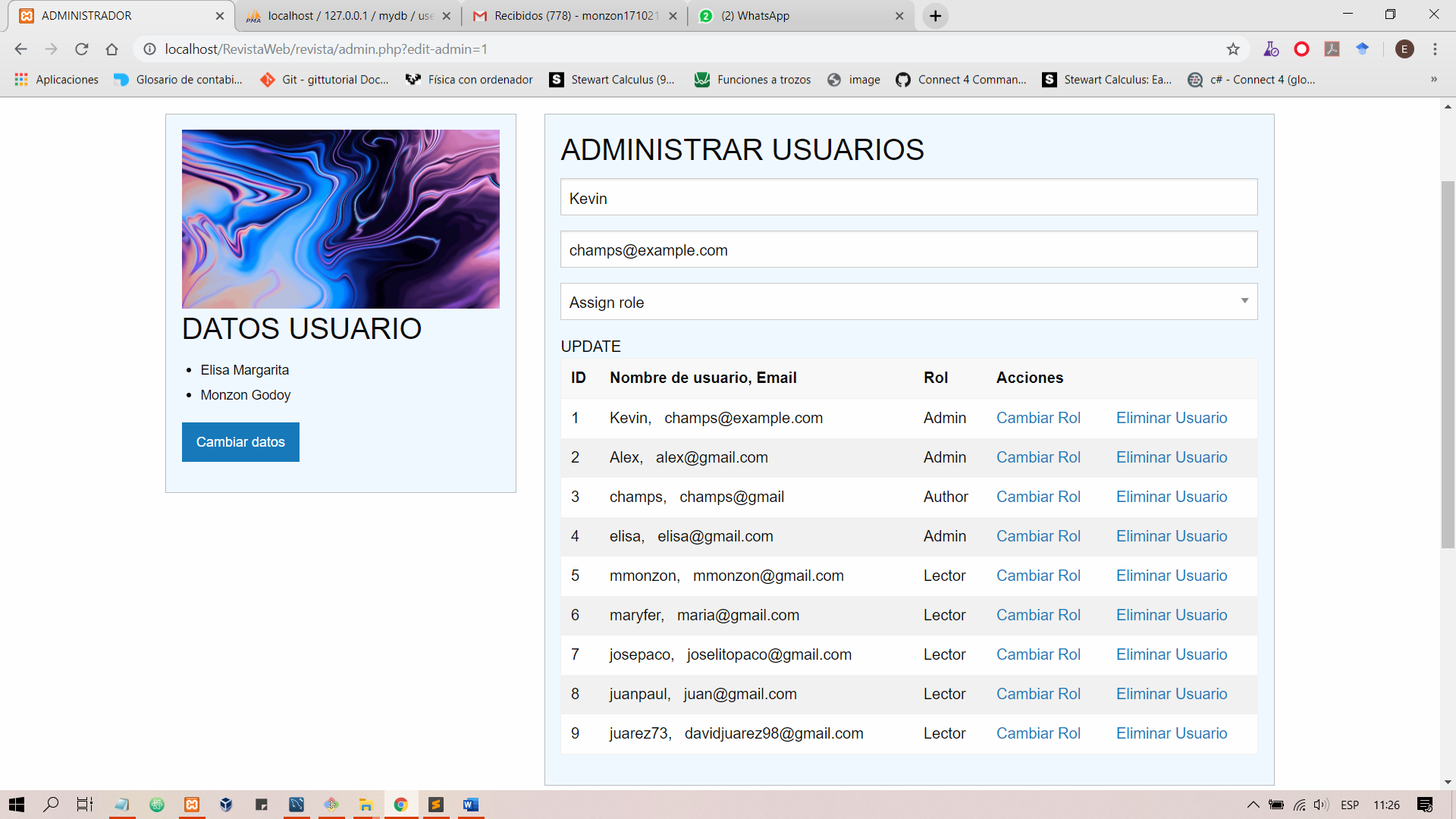


Es importante tomar en cuenta que la contraseña se debe cifrar utilizando md5 de lo contrario al momento de ser decodificada en el inicio de sesión, no se podrá ingresar

## Administración de usuarios en general

Dentro de las funciones del administrador se encuentra la función de administrar usuarios, esto se lleva acabo en el perfil del admin.

Al dar clic en cambiar rol se despliega la información del usuario a administrar, asimismo se puede eliminar



El código funcional es el siguiente:

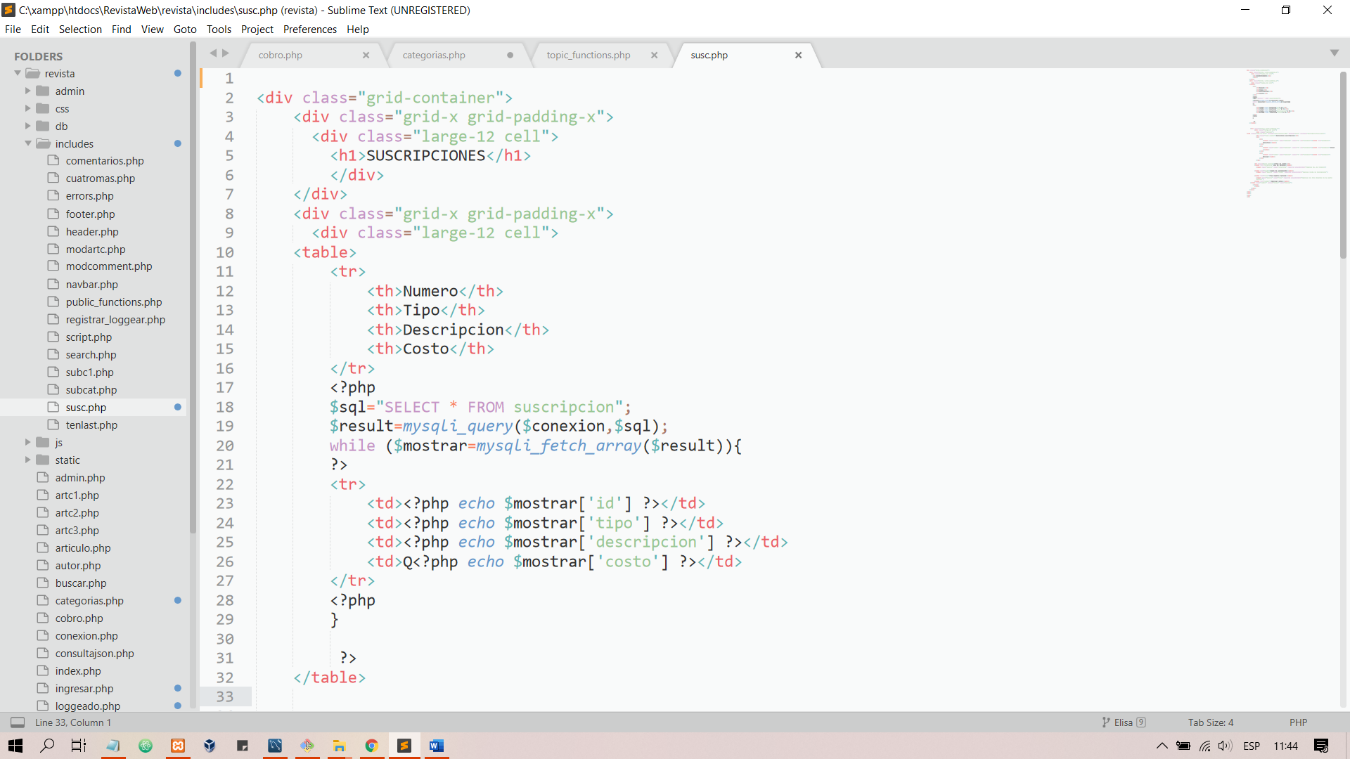
## Creación de artículos

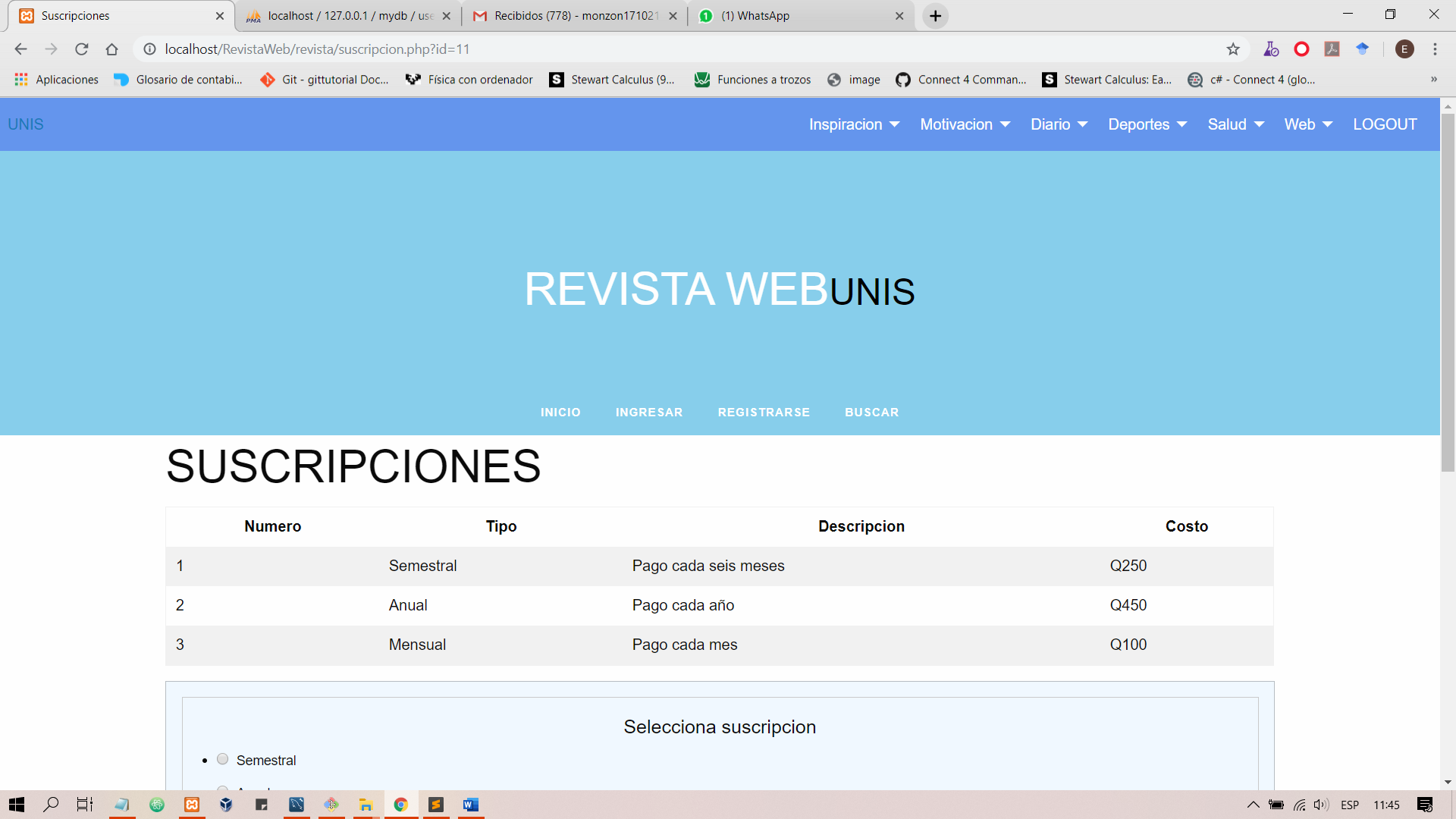
## Categorías

## Subcategorías

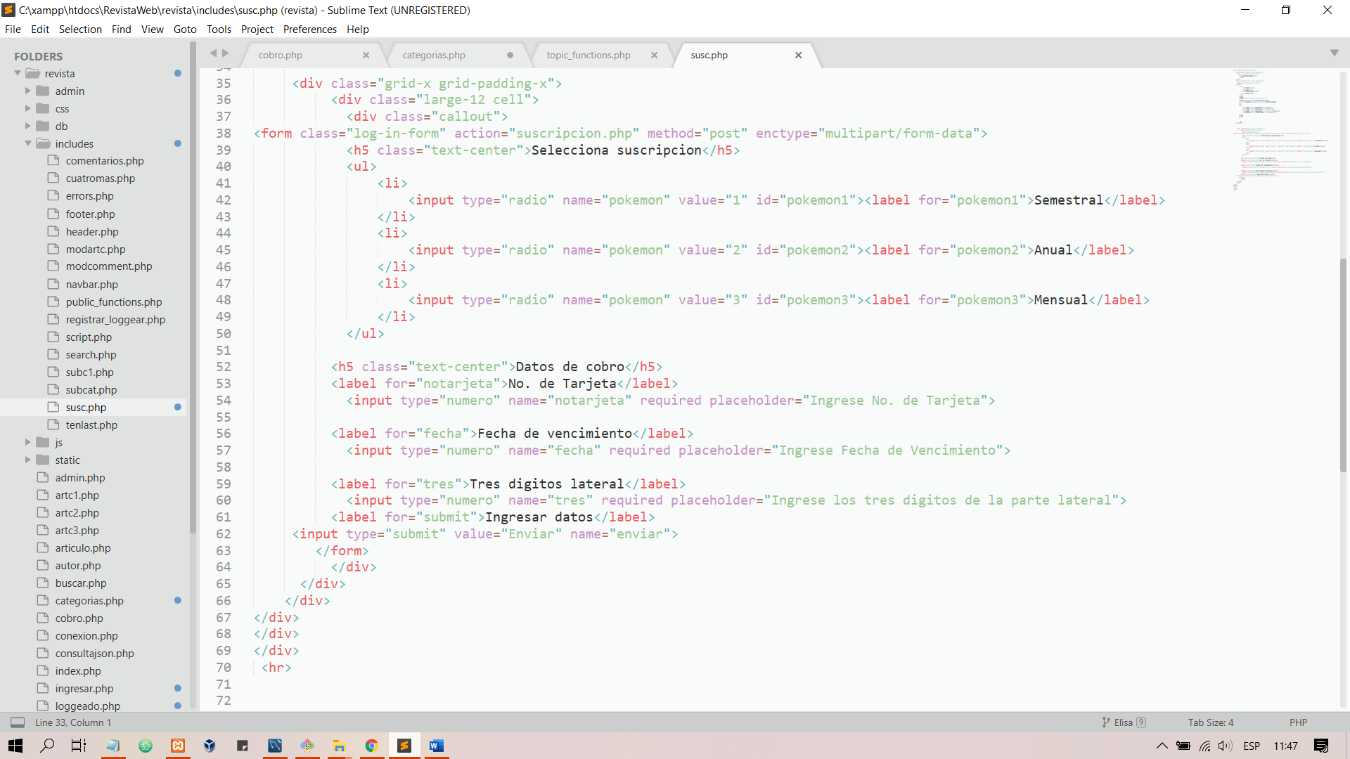
## Venta suscripciones

En primera instancia el usuario logueado podrá ver las suscripciones que sean añadidas de manera dinámica





En el perfil del usuario lector tiene la opción de adquirir o no una suscripción, la cual está ligado a la funcionalidad de paga.

En este formulario se selecciona el tipo de membresia y se debe ingresar obligatoriamente los datos de la tarjeta

Posteriormente los datos serán visibles en el perfil del usuario

## Búsqueda con filtros y ordenamiento

### Actualización de articulo

# Funcionalidades MySQL

## Procedimiento con parámetros out

## Vistas