TECNOLOGICO DE ESTUDIOS SUPERIORES DE ECATEPEC

BASE DE DATOS DISTRIBUIDAS

ISC. FRANCISCO JAVIER HERNANDEZ BARAJAS

PROYECTO SEGUNDO DEPARTAMENTAL

GRUPO: 5701

INTEGRANTES:

BAUTISTA GARCIA LUIS ENRIQUE  
GONZALEZ MENDEZ MARIA DOLORES  
GUERRA VELAZQUEZ JESSICA  
LOPEZ DANIEL CARLOS ROBERTO  
MAYA REYES ABIGAIL  
MERA MOJARAZ NANCY BERENICE  
NAVA TERRONES PERLA GUADALUPE  
OROZCO GUTIERREZ RICARDO  
QUEBRADO SERVIN GERARDO  
RAMIREZ LEON EDER JOSSIMAR  
REYES ALCIVAR LUIS ANGEL

# INDICE

OBJETIVO. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

INTRODUCCION. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

DESARROLLO. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

Interfaces de Pagina web. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

Código socket cliente. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

Código socket servidor. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

CONCLUSION. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

REFERENCIAS. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

# OBJETIVO

Por medio de este proyecto se pretende poder llevar a cabo el desarrollo de réplicas para la base de datos de un Gimnasio haciendo uso de las herramientas de sockets.

# INTRODUCCIÓN

Por medio de este proyecto se desea poder implementar sockets en php y Python, switcheo de base de datos y servicio de aplicaciones web, servidor de correos de igual forma se llevara a cabo el servicio de una aplicación web, el cual estará conectada a una base de datos y se estará llevando a cabo el desarrollo de sockets en php y python para poder llevar a cabo la comunicación de la aplicación con los servicios en una red dentro del sistema el cual estará definido como el punto de conexión.

Por medio de un socket quedara definido por un par de [direcciones IP](https://www.ecured.cu/Protocolo_de_Internet) local y remota, un protocolo de transporte y un par de números de puerto local y remoto. Para que los dos programas puedan comunicarse, en este caso la aplicación con el servidor web.

Un servidor de socket es un servicio asignado a un puerto particular que escucha las solicitudes entrantes y las responde.  Los servidores de socket normalmente se ejecutan continuamente como un servicio o un daemon del sistema.

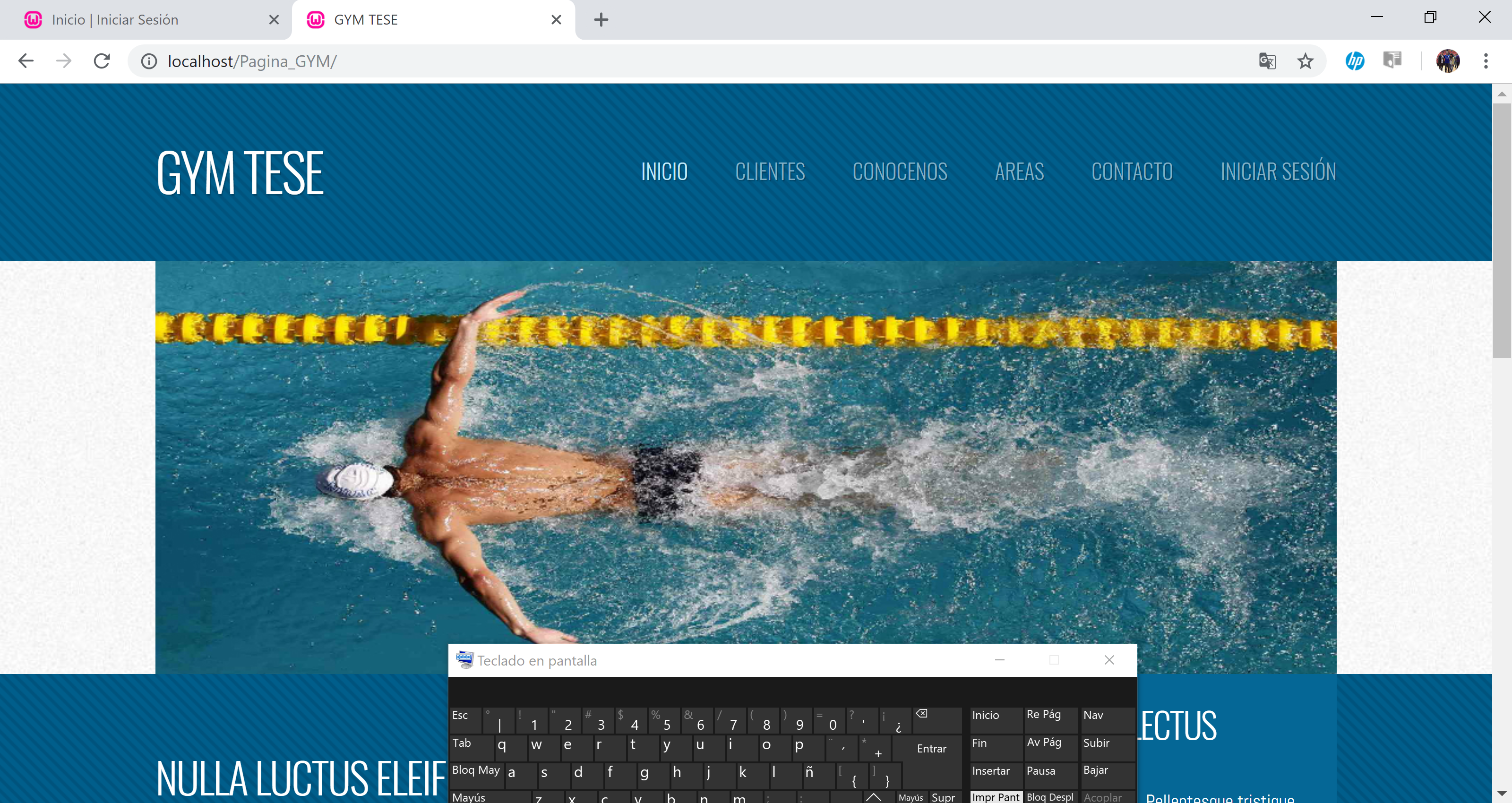
El modelo más básico de los sockets consta de 2 simples programas, un servidor y un cliente.Básicamente el programa servidor comienza a “escuchar” en un puerto determinado (nosotros lo especificamos), y posteriormente el programa que la hace de “cliente” debe conocer la ip o nombre de dominio/hostname del servidor y el puerto que está escuchando, al saber esto simplemente solicita establecer una conexión con el servidor.

# DESARROLLO

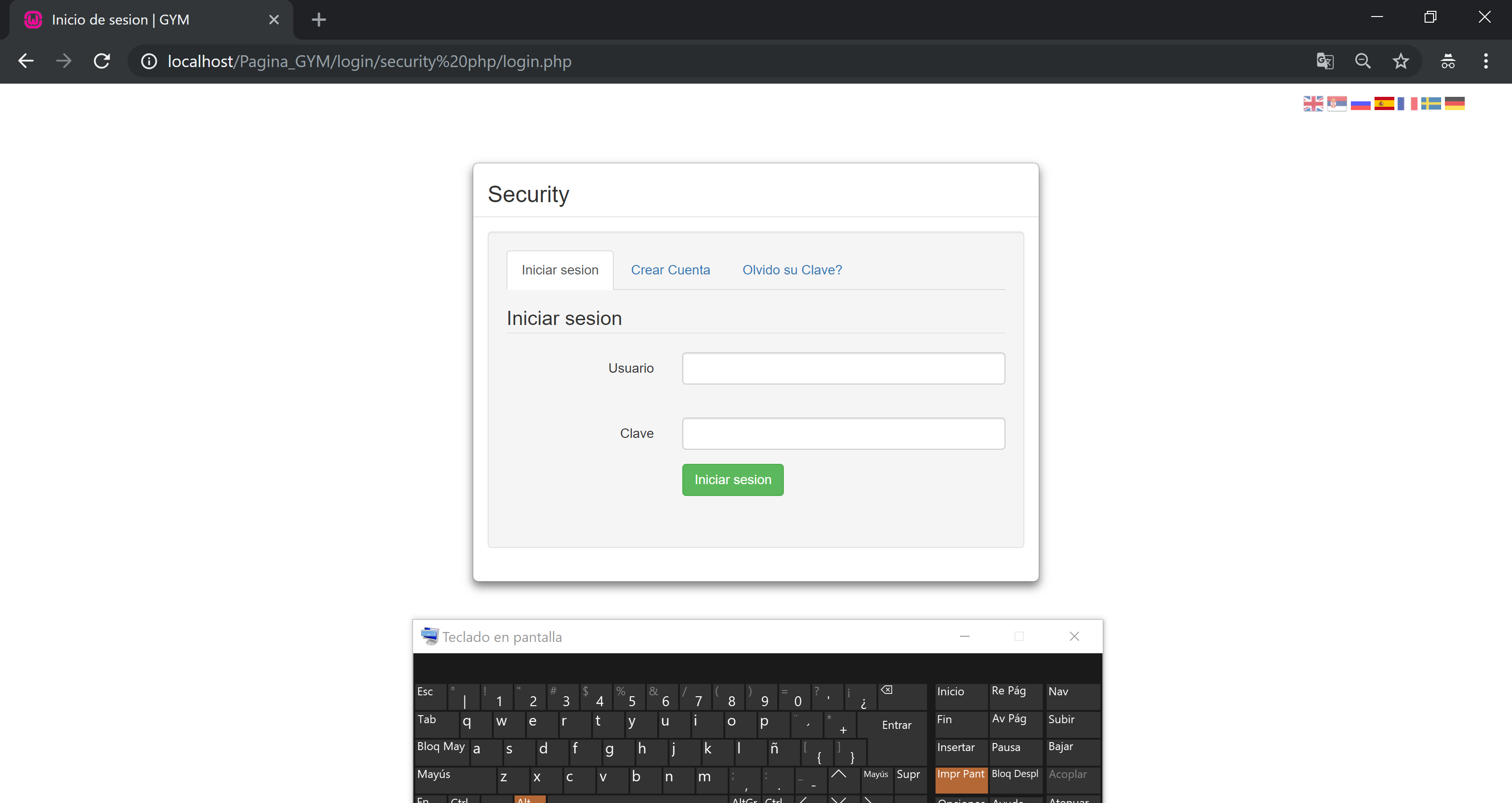
Página de inicio de GYM

Cambios realzados

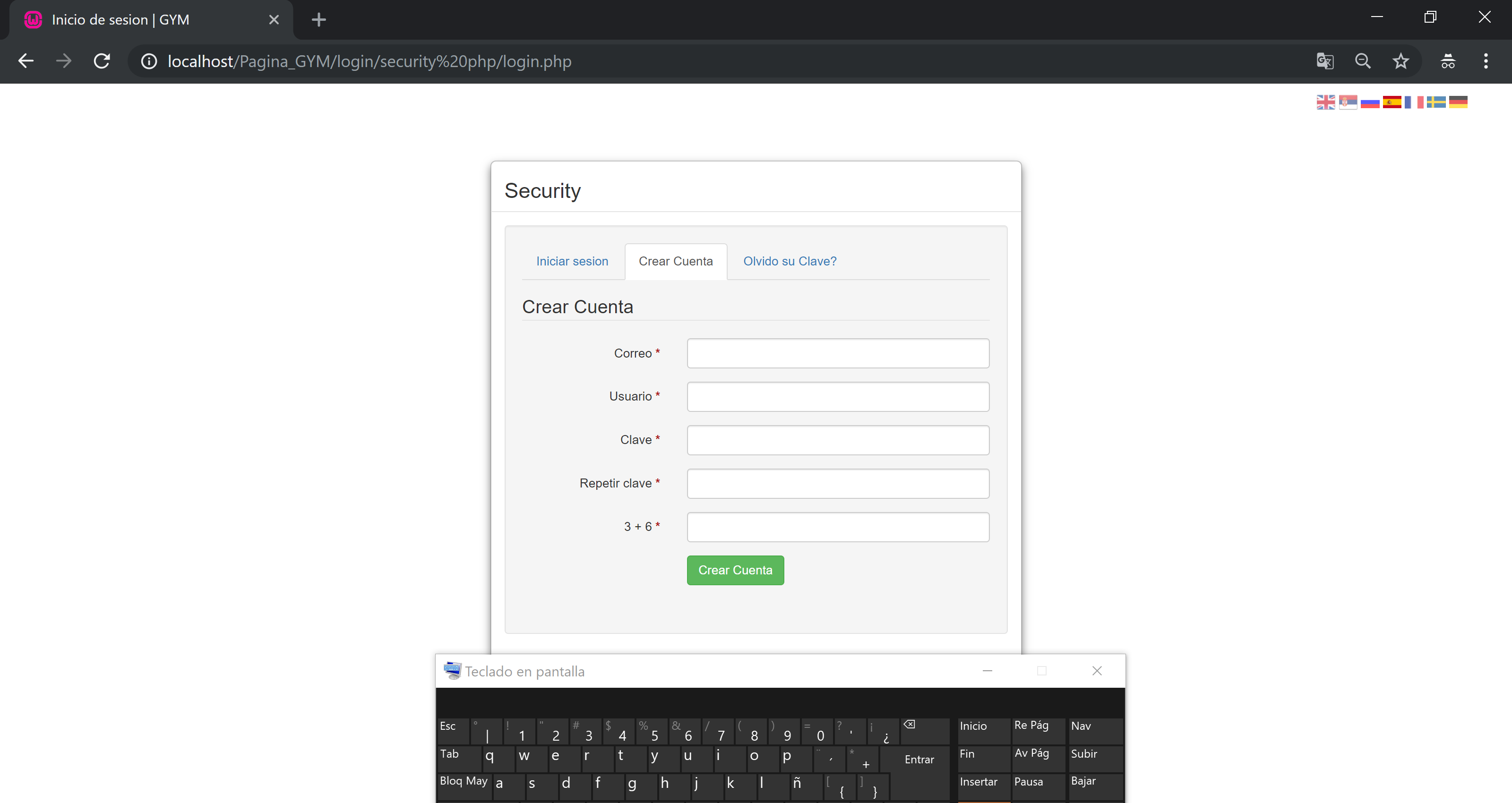
* Respectivos enlaces
* Ligas de botón



En el apartado de la página se muestra el enlace de iniciar sesión el cual nos redirige al login del sistema, dentro de la cual el usuario deberá de contar ya con un una cuenta en caso contrario el usuario podrá darse de alta en la pestaña crear cuenta.



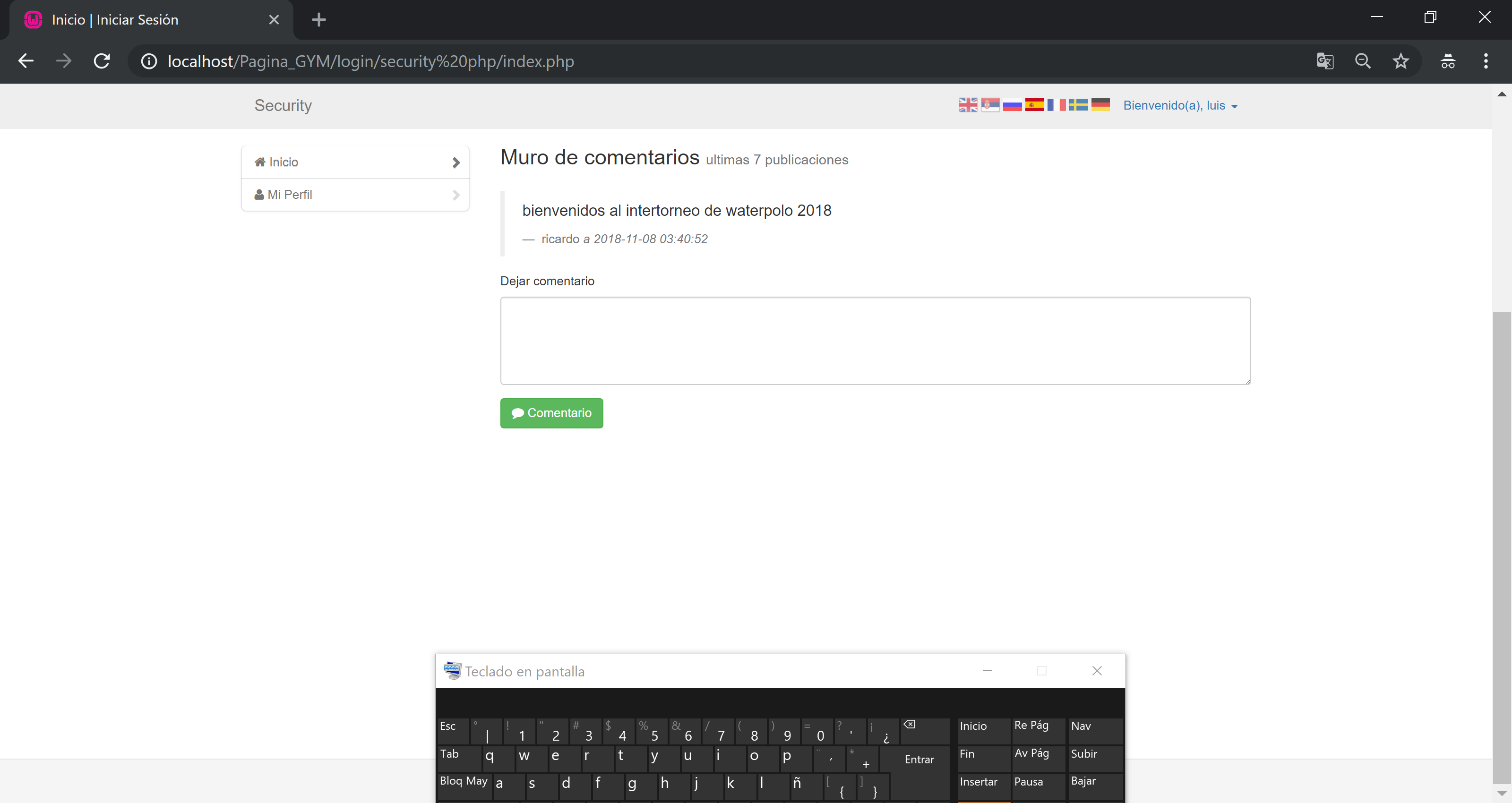
En la página se muestra la pestaña crear cuenta del cual los usuarios pueden darse de alta en el sistema.



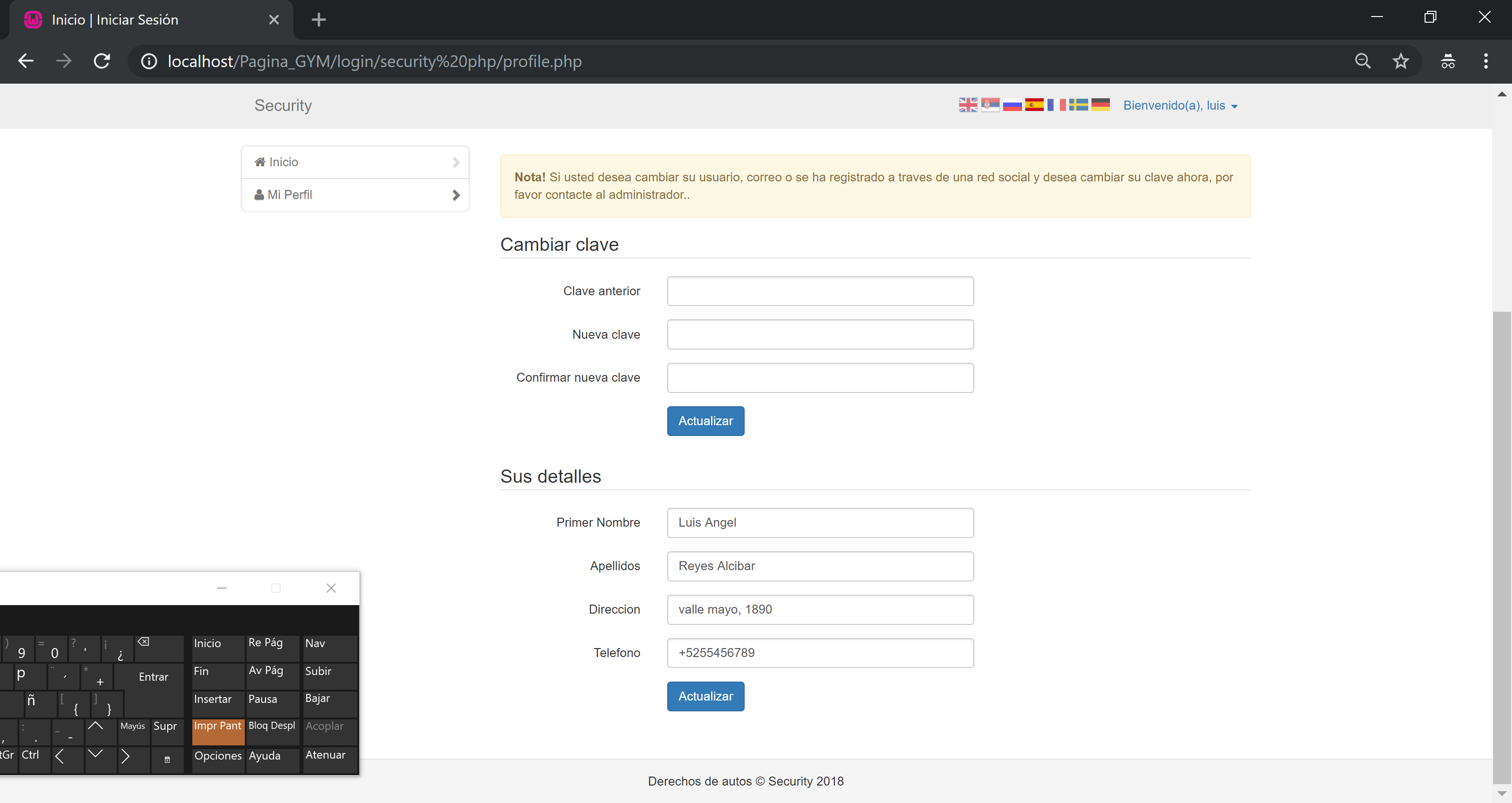
En la pagina se muestra la pestaña Olvido Contraseña, el cual permite al usuario recuperar la contraseña del mismo.



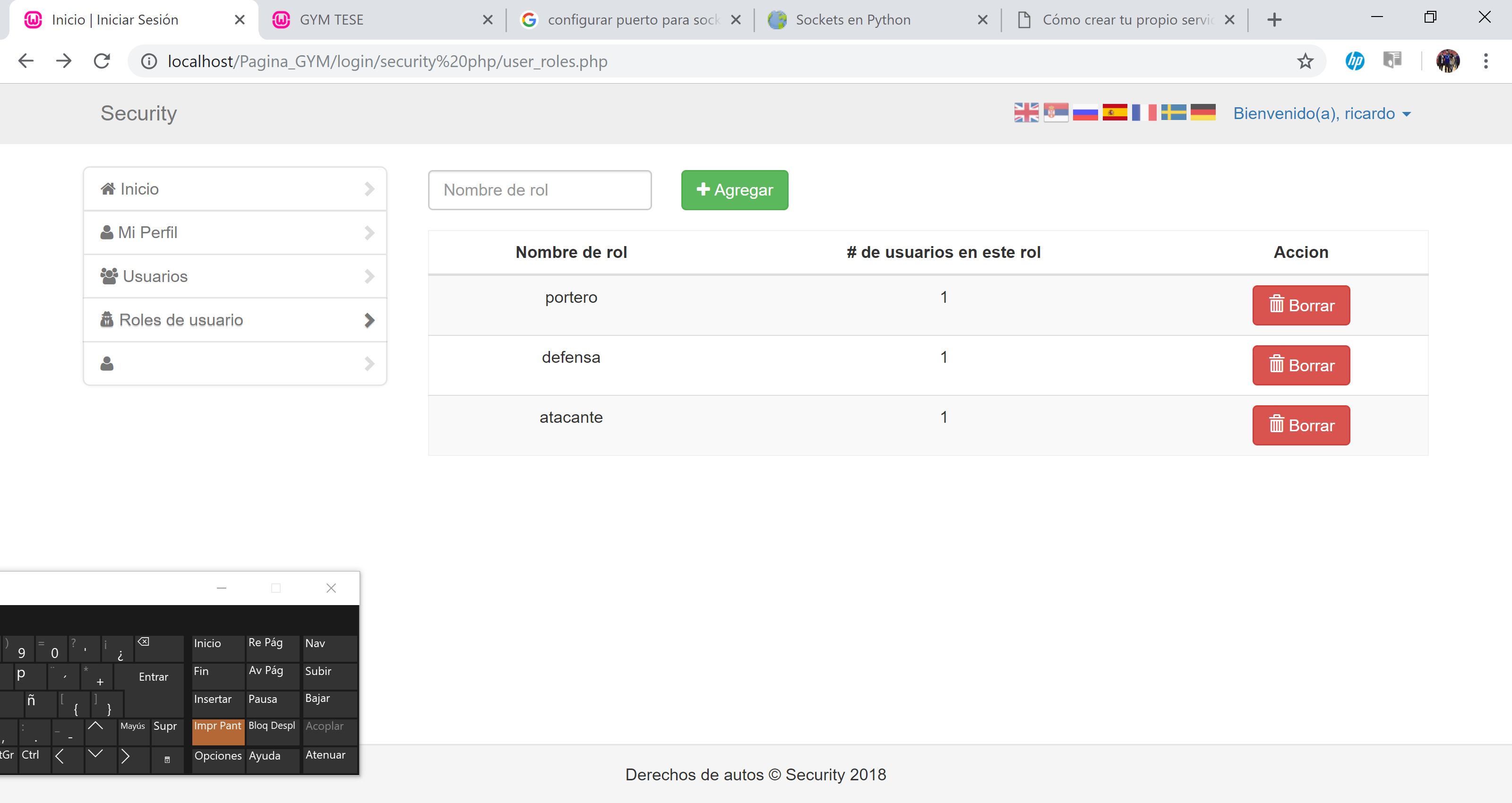
En esta ventana se muestra la pantalla de bienvenida de waterpolo al usuario, dentro de la misma existe un campo del cual se podrán dejar comentarios del evento.



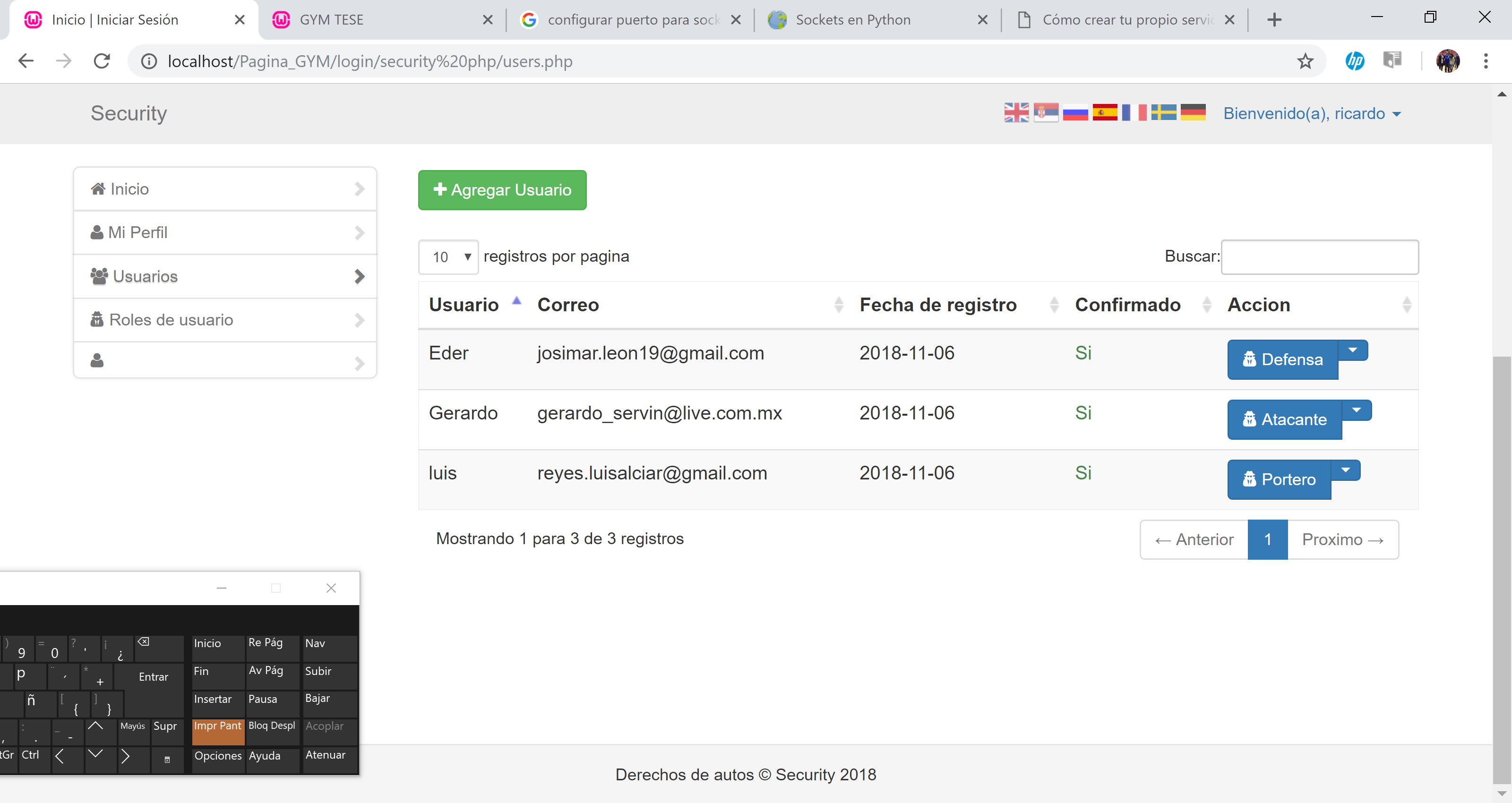
Dentro de la ventana “Mi Perfil” existen la opción de actualizar los datos del usuario y el cambio de su contraseña y entre otros datos.



En esta parte el usuario tiene la opción de agregar los roles que van a desempeñar cada uno de los participantes del evento.

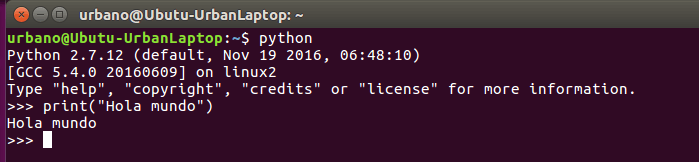


Podremos observar los usuarios, correo, fecha de registro, estado de confirmación y la acción que tendrá cada uno de los usuarios.



Estaremos creando un servidor en Python.

Lo primero que debemos tener, evidentemente, es Python instalado. Abre una terminal y escribe “Python” y te aparecerá información acerca de la versión que tienes instalada. Podrás ejecutar código en tiempo real cuando la herramienta esté en ejecución.

[](http://elfreneticoinformatico.com/wp-content/uploads/2017/11/C1.png)

Para crear comunicación entre dos programas utilizaremos sockets. Los sockets son un concepto abstracto. Con ellos dos programas pueden comunicarse. Estos programas pueden estar en la misma máquina o bien ejecutarse en dispositivos diferentes.

## CODIGO SOCKET CLIENTE

#!/usr/bin/python

# -\*- coding: utf-8 -\*-

# Ejemplo cliente - servidor en python

# Programa Cliente

# www.elfreneticoinformatico.com

import socket #utilidades de red y conexion

#declaramos las variables

#ipServidor = "127.0.0.1" #es lo mismo que "localhost" o "0.0.0.0"

ipServidor = "192.168.0.102"

puertoServidor = 5000

#Configuramos los datos para conectarnos con el servidor

#socket.AF\_INET para indicar que utilizaremos Ipv4

#socket.SOCK\_STREAM para utilizar TCP/IP (no udp)

#Estos protocolos deben ser los mismos que en el servidor

cliente = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

cliente.connect((ipServidor, puertoServidor))

print("Conectado con el servidor ---> %s:%s" %(ipServidor, puertoServidor))

while True:

msg = raw\_input("> ")

cliente.send(msg)

respuesta = cliente.recv(4096)

print(respuesta)

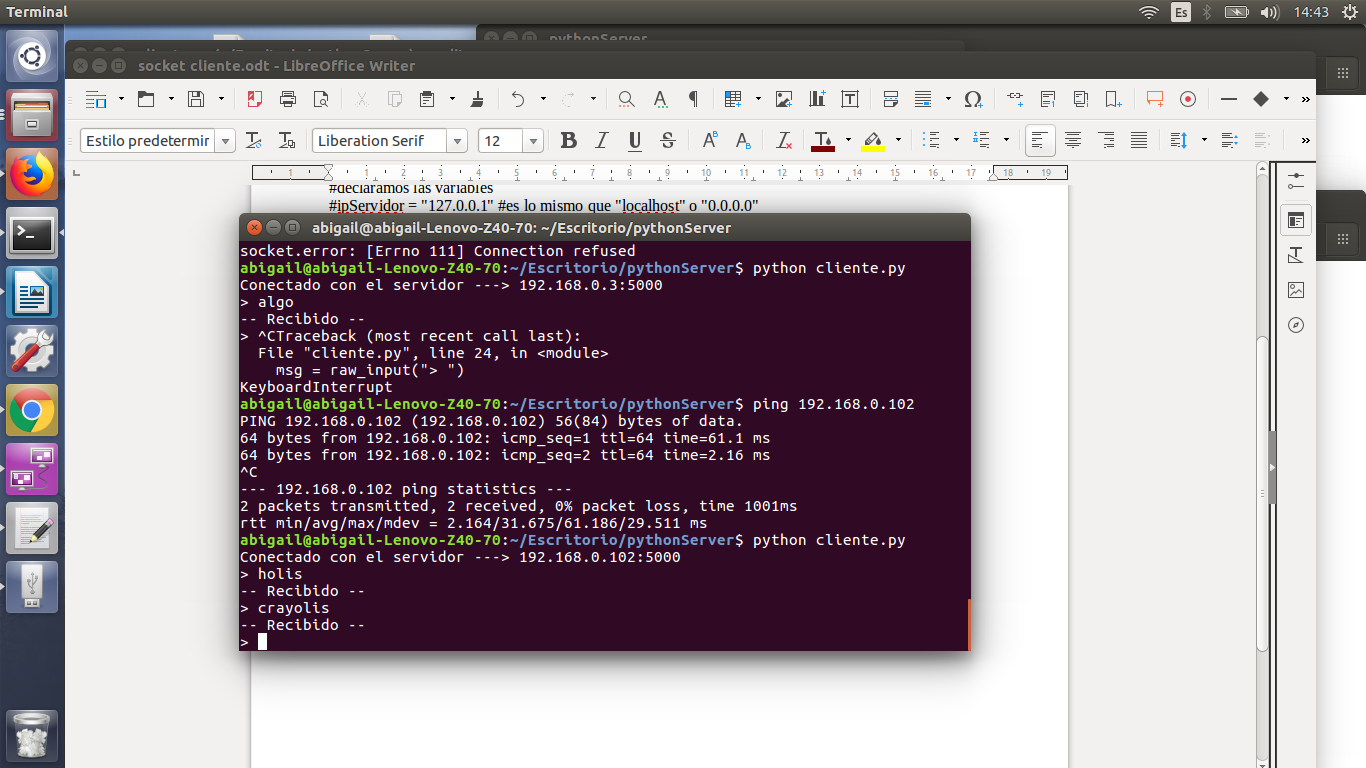
if respuesta == "exit":

break;

print("------- CONEXIÓN CERRADA ---------")

cliente.close()

Nuestro servidor ya está terminado. Ahora, debemos crear el cliente que se conectará con él.



## CODIGO SOCKET SERVIDOR

#!/usr/bin/python

# -\*- coding: utf-8 -\*-

# Programa Servidor

# www.pythondiario.com

import socket

import sys

# Creando el socket TCP/IP

sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

# Enlace de socket y puerto

server\_address = ('192.168.0.100', 9797)

print>>sys.stderr,'empezando a levantar %s puerto %s' % server\_address sock.bind(server\_address)

# Escuchando conexiones entrantes

sock.listen(1)

while True:

# Esperando conexión

print >>sys.stderr, 'Esperando para conectarse' connection, client\_address = sock.accept()

while True:

# Esperando conexion

print >>sys.stderr, 'Esperando para conectarse'

connection, client\_address = sock.accept()

try:

print >>sys.stderr, 'concexion desde', client\_address

# Recibe los datos en trozos y reetransmite

while True:

data = connection.recv(19)

print >>sys.stderr, 'recibido "%s"' % data

if data:

print >>sys.stderr, 'enviando mensaje de vuelta al cliente'

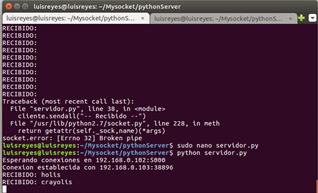
connection.sendall(data)

else:

print >>sys.stderr, 'no hay mas datos', client\_address break finally:

# Cerrando conexion

connection.close()

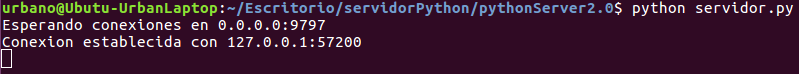


Solo nos queda ejecutar ambos programas y probarlos.

Primero arrancamos el servidor (python servidor.py)  y se quedará a la espera de conexión:

[http://elfreneticoinformatico.com/wp-content/uploads/2017/11/C2.png](http://elfreneticoinformatico.com/wp-content/uploads/2017/11/C2.png)A continuación, arrancamos el cliente (python cliente.py):

[http://elfreneticoinformatico.com/wp-content/uploads/2017/11/C3.png](http://elfreneticoinformatico.com/wp-content/uploads/2017/11/C3.png)Y veremos que el servidor ya ha visto también la conexión:

[](http://elfreneticoinformatico.com/wp-content/uploads/2017/11/C4.png)Lo que debemos hacer ahora es escribir cualquier mensaje en el cliente.

# CONCLUSION

Por medio de este proyecto aprendimos a realizar replicas en bases de datos haciendo uso de sockets tanto para el servicio del cliente y el servidor, de igual forma realizamos inserciones de datos para poder llevar a cabo las consultas realizadas dentro de la base.

Utilizamos y configuramos el servidor en el lenguaje de Python y de igual forma pudimos hacer la administración de usuario dentro del sistema así como dar de altas y bajas a cada uno para que puedan tener acceso dentro del sistema.

# REFERENCIAS

https://www.google.com.mx/search?q=socket+en+python&rlz=1C1CHBF\_esMX814MX814&oq=socket&aqs=chrome.2.69i57j69i59l5.4176j0j7&sourceid=chrome&ie=UTF-8

<https://docs.microsoft.com/es-es/sql/database-engine/database-mirroring/database-mirroring-and-replication-sql-server>

<https://realpython.com/python-sockets/>

http://recursostic.educacion.es/observatorio/web/eu/software/servidores/176-crear-una-replica-de-la-base-de-datos