



UNAH

UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE
INGENIERÍA EN SISTEMAS

IS912 - Desarrollo Web
Ejercicios Prácticos

Catedrático: Erick Vladimir Marín



UNAH

UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

IS-912, Desarrollo Web

Ejercicios Prácticos

Catedrático: Ing. Erick Vladimir Reyes Marín

Última actualización: 20/Junio/2018



UNAH

UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE
INGENIERÍA EN SISTEMAS

IS912 - Desarrollo Web
Ejercicios Prácticos

Catedrático: Erick Vladimir Marín



Contenido

| | |
|------------------|----|
| Nota..... | 3 |
| Ejercicio 1..... | 4 |
| Ejercicio 2..... | 6 |
| Ejercicio 3..... | 8 |
| Ejercicio 4..... | 10 |
| Ejercicio 6..... | 12 |



UNAH

UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE
INGENIERÍA EN SISTEMAS

IS912 - Desarrollo Web
Ejercicios Prácticos

Catedrático: Erick Vladimir Marín

Nota

Los ejercicios propuestos en este documento son con fines educativos para estudiantes de la Universidad Nacional Autónoma de Honduras, queda prohibida su reproducción parcial o total del mismo o su uso en evaluaciones de otros cursos ajenos a los impartidos por él autor. Cada ejercicio necesita de un proyecto base para poder desarrollarlo, estos podrán encontrarlos en el siguiente repositorio de github:

<https://github.com/UNAH-IS/IS912-BaseEjercicios>

Ejercicio 1

El objetivo es desarrollar un pequeño chat básico. Se le provee un proyecto web con una plantilla similar a whatsapp tal y como lo muestra la Fig. 2.

1. Como primer paso debe cambiar la apariencia total de dicha plantilla para que su interfaz sea similar a Telegram (Fig. 3).
2. Mediante javascript agregar la funcionalidad que permita que al hacer click sobre el botón enviar se agregue el mensaje de la caja de texto como un nuevo mensaje en la región que muestra la conversación, es decir deberá lucir como un mensaje enviado.
3. Utilizar AJAX para enviar la información del nuevo mensaje al servidor, esta información la recibirá una rutina en PHP que almacenará los mensajes a un archivo de texto. La información que enviará será:
 - a. Usuario seleccionado o usuario emisor. (Se obtiene de la lista desplegable, en el screenshot se muestra vegueta).
 - b. Usuario receptor. Este se obtendrá de la lista de la sección izquierda.
 - c. Mensaje a enviar. Este se obtendrá de la caja de texto.

*Nota: En la Fig. 1 se muestra el flujo de información entre el Frontend y el Backend.

4. Crear una nueva rutina en Javascript que esté haciendo peticiones Ajax cada 30 segundos en busca de más mensajes, esta petición Ajax solicitara a una rutina PHP la lectura del archivo de mensajes y los mostrara en pantalla.

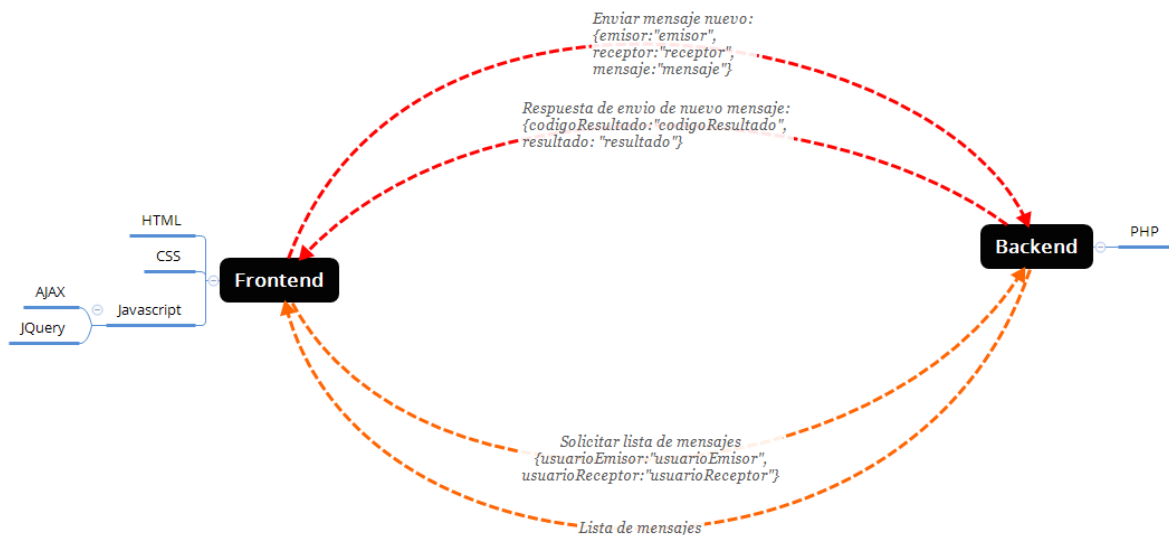


Fig. 1

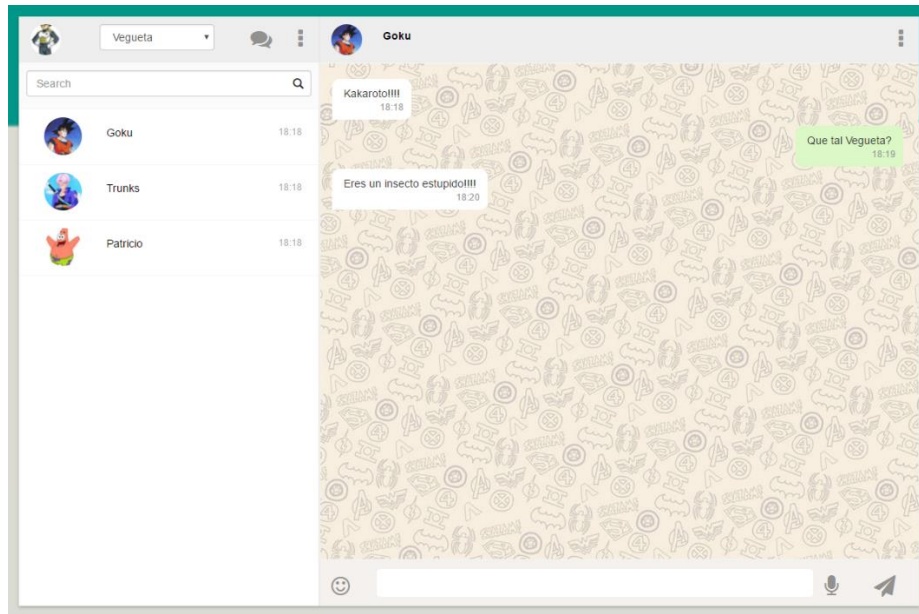


Fig. 2

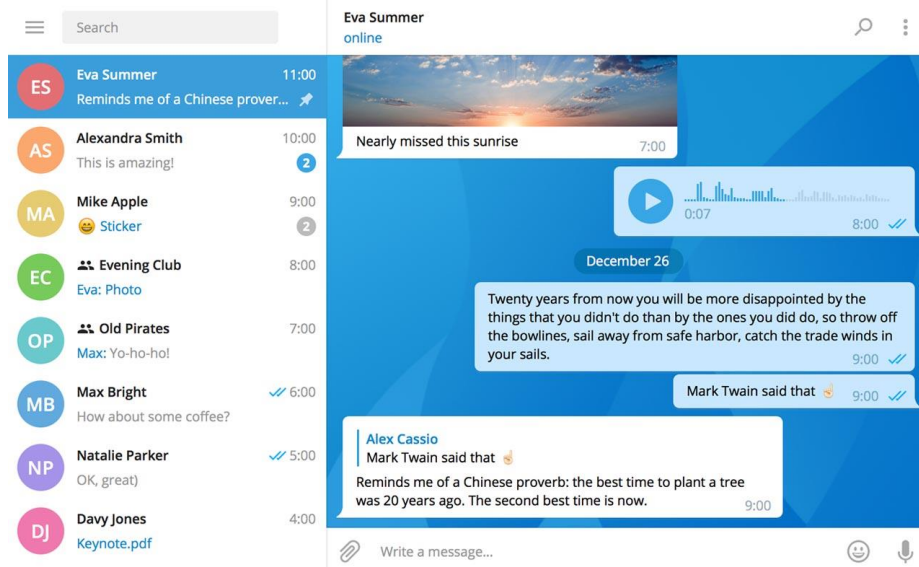


Fig. 3

Ejercicio 2

Proyecto base

Lenguajes/Tecnologías

PHP, HTML, CSS, JS, AJAX, Bootstrap, Gestión de archivos.

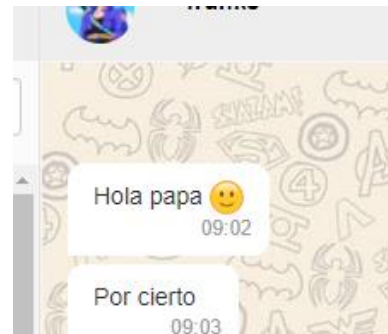
Antes de comenzar:

- Copie la carpeta llamada fbmessenger en la dirección c:\wamp\www
- Acceda desde un navegador web a la dirección http://localhost/fbmessenger

Requisitos funcionales:

1. Al momento de guardar un mensaje se debe verificar si en la cadena existen ciertos conjuntos de caracteres o shortcuts que hacen referencia a imágenes llamadas emojis, dependiendo del shortcut escrito el texto debe ser sustituido por la etiqueta html correspondiente para mostrar una imagen. Las sustituciones a realizar son las siguientes:

| Shortcut | Emoji | Ruta imagen |
|----------|-------|-----------------------|
| :) | 😊 | img/emojis/emoji1.png |
| XD | 😄 | img/emojis/emoji2.png |
| :P | 😜 | img/emojis/emoji3.png |
| :(| 😞 | img/emojis/emoji4.png |
| :* | 😘 | img/emojis/emoji5.png |
| X_X | 😵 | img/emojis/emoji6.png |
| ** | 🐶 | img/emojis/emoji7.png |



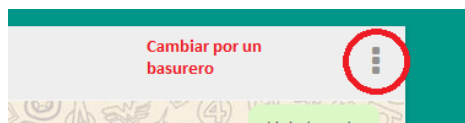
Nota:

*Para reemplazar una cadena con otra desde PHP utilizar la función str_replace:

https://www.w3schools.com/Php/func_string_str_replace.asp

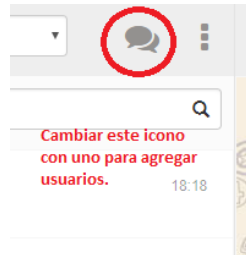
2. Sustituir el icono de la esquina superior derecha con el icono de un basurero, al dar click sobre este botón deberá eliminar la conversación entre los usuarios seleccionados. Utilizar la función de PHP unlink()

https://www.w3schools.com/php/func_filesystem_unlink.asp

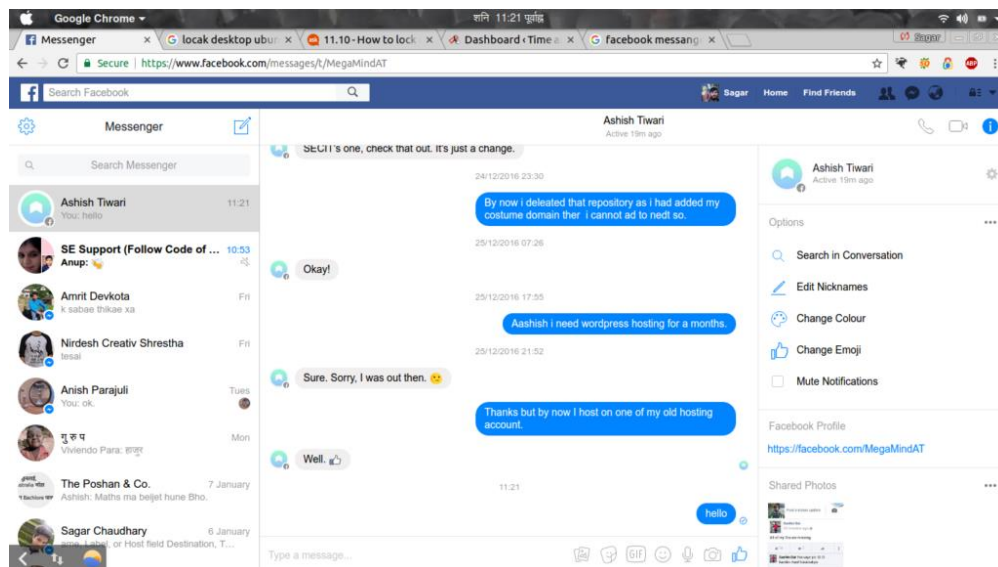


Puede utilizar un icono de la librería fontawesome, un glyphicon o una imagen.

3. Cambiar el icono indicado en la siguiente imagen por uno más adecuado. Dicho botón está configurado para abrir una ventana modal de Bootstrap, cree un formulario dentro de la ventana modal para capturar el código del usuario, el nombre del usuario y un select list con las imágenes de perfil que están disponibles en la carpeta img/profile-pics. Dar la funcionalidad para guardar la información en el archivo de usuarios.csv. Cada vez que se guarde un nuevo usuario se debe refrescar las listas de contactos.



4. Cambiar la apariencia del proyecto a algo más parecido a la siguiente imagen. Cambiar colores y favicon (buscar uno de facebook).



Ejercicio 3

Se adjunta un archivo comprimido con un proyecto en PHP,HTML,CSS y JS, también hace uso de las versiones más recientes de Bootstrap y Fontawesome. Descomprimir el archivo dentro de la carpeta principal de Wamp server o el servidor que tenga instalado en su computadora. Se recomienda leer los archivos principales index.php y controlador.js, estudie detenidamente el funcionamiento actual del proyecto.

El proyecto que se le provee busca imitar la funcionalidad básica de la plataforma de gestión de archivos en línea llamada Dropbox.

Una vez copiado el proyecto acceder a la siguiente dirección (en caso de descomprimirse utilizando otra ruta de carpetas considerarlas en la URL):

<http://localhost/dropbocs/index.php?carpeta=home>

Como puede observar en la URL se incluye el parámetro en GET que se llama carpeta=home, el proyecto ya está programado para recuperar este valor e incluirlo en un input(inactivo) con el id txt-carpeta-actual el cual se refleja de la siguiente forma:

home

En la parte inferior se encuentra una tabla html simulando el contenido de una carpeta de archivos, dicha información está escrita estáticamente utilizando html, su primera misión es generar dicha tabla dinámicamente utilizando una petición AJAX obteniendo la información de un archivo con extensión csv con el mismo nombre que figura dentro del input mencionado en el párrafo anterior. Por ejemplo, si el nombre de la carpeta del input txt-carpeta-actual esta el valor **home** se debe consultar el archivo que esta en data/home.csv, si el valor es home/carpeta1 se debe generar la tabla a partir del archivo data/home/carpeta1.csv y así respectivamente para cada caso. En el caso de las **carpetas si se visualizan también físicamente como carpetas**, en el caso de los **archivos no**, únicamente se listan o se registran dentro del csv.

La estructura de todos los archivos que gestionan esta información tiene el siguiente formato:

nombreElemento,tipo,fechaModificacion,fechaCreacion,usuario,tamaño

nombreElemento: El nombre de la carpeta o archivo

tipo: file en caso de ser un archivo y folder en caso de ser carpeta

fechaModificacion: Registro de la última de modificación del archivo.

fechaCreacion: Registro de la creación del archivo.

usuario: Nombre del usuario que registro el archivo.

tamaño: Tamaño registrado del archivo.

Preste especial atención en el html estático que genera la tabla actual, en cada caso utiliza iconos del framework Fontawesome, incluir los iconos dependiendo del tipo de registro que se va a imprimir, en caso de ser una carpeta incluir el icono del folder y en caso de ser archivo verificar la extensión que esta registrada e imprimir el icono correspondiente. Utilizar los iconos de las diferentes extensiones que se muestran de ejemplo.

Observe que las **etiquetas html** para carpetas son diferentes que en el caso de los archivos.

En el caso de las carpetas son **anchors** con el formato index.php?carpeta=NOMBRE-CARPETA, esto ya esta programado para que cargue de nuevo la pagina y envíe via GET la información para mostrarla en el input mencionado en el primer párrafo (Puede hacer pruebas dando click y observar los cambios).



En el caso de los archivos son botones que ejecutan una función javascript que muestra una ventana modal para mostrar el detalle del archivo seleccionado, envíe en esa función todos los parámetros para visualizarlos en la ventana modal. Para esto no es necesario hacer una petición AJAX porque podría poner toda la información de parámetro (esto porque es poca información).

Al final hay dos botones, uno para crear carpetas y otro para crear archivos, ambos abren una ventana modal para crear registros de dichos elementos. Para las carpetas hay que guardar registros con el siguiente formato

nombreCarpeta,**folder**,fechaModificacion,fechaCreacion,usuario,tamaño

Además de registrar dicha información se debe crear una carpeta vacía utilizando PHP, para ello utilizar una función llamada mkdir. Ejemplo: mkdir("/direccion/del/directorio", 0700); El numero 0700 incluirlo únicamente si trabaja en Linux, tiene que ver con los permisos de la carpeta creada.

Para los archivos solo basta con crear el registro en el archivo csv correspondiente. El formato del registro a guardar es:

nombreArchivo,**file**,fechaModificacion,fechaCreacion,usuario,tamaño

Tanto los registros de archivos y carpetas deben almacenarse en un archivo con el nombre de la carpeta principal a la que pertenece, ese valor lo obtendrá del input txt-carpeta-actual.

***Nota:** Para todas las funcionalidades utilizar AJAX, Ante cualquier consulta Google y Stackoverflow están hoy de su lado, todo saldrá bien, lea detenidamente el enunciado y los archivos proveídos.



Ejercicio 4

Para esta evaluación se adjunta un proyecto de NodeJS llamado “trello”, en dicho proyecto ya están instalados los módulos express, mysql y body-parser. Hay una carpeta llamada public donde están los archivos estáticos, en ella el archivo index.html (fig. 1) debe ser el archivo que se debe desplegar cuando se intente acceder a la dirección raíz (http://localhost:puerto/).

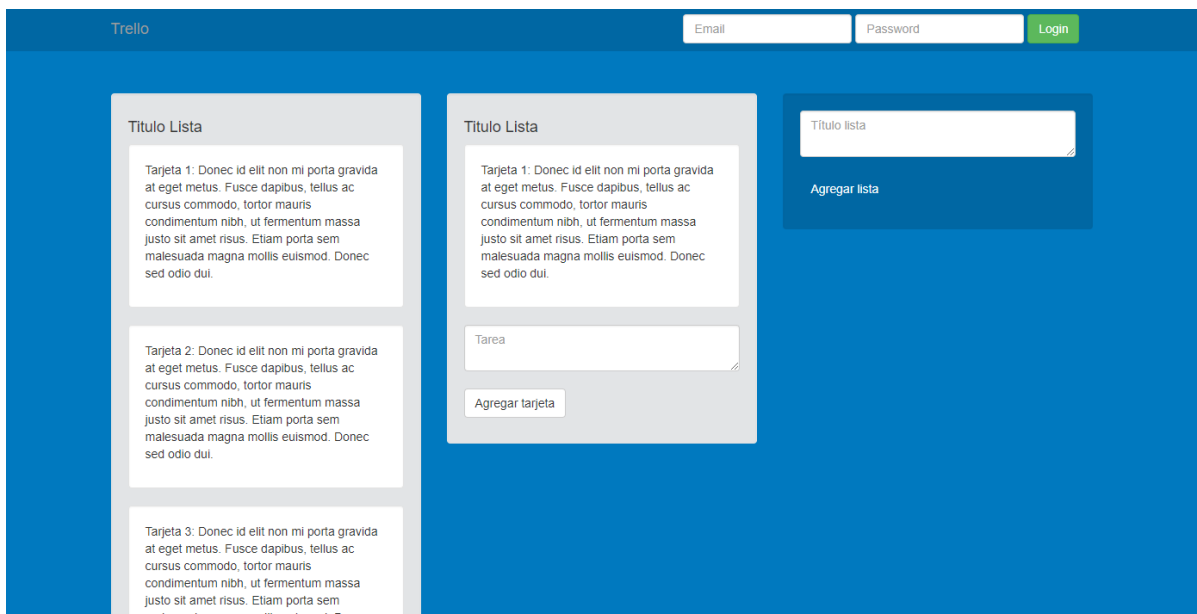


Fig. 1

1. Cree una base de datos en mysql con la siguiente estructura:
 - Usuarios
 - codigo_usuario
 - nombre_usuario
 - contrasena
 - Listas
 - codigo_lista
 - titulo_lista
 - codigo_usuario
 - Tarjetas
 - codigo_tarjeta
 - nombre_tarjeta
2. Agregar funcionalidad al formulario de login de la esquina superior derecha, lo que debe hacer es una petición AJAX enviar el usuario y la contraseña y verificar si existe un usuario con esos datos, en caso de ser así retornar su nombre y todas las listas relacionadas a ese usuario para generar todas las listas con sus respectivas tareas, en el archivo index.html se muestra la estructura de dos listas con tareas de ejemplo para que pueda generar la estructura.
3. Dar funcionalidad para agregar una nueva tarea dependiendo de la lista que este ubicado.
4. Dar funcionalidad para agregar nuevas listas.



UNAH

UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE
INGENIERÍA EN SISTEMAS

IS912 - Desarrollo Web
Ejercicios Prácticos

Catedrático: Erick Vladimir Marín

Para todas las peticiones utilizar respuestas en formato JSON, ya hay un archivo llamado controlador.js con los eventos para los botones.

Adjuntar el proyecto completo con un export(.sql) de la base de datos creada.

LU
CEM
ASPI
CIO



UNAH

UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE
INGENIERÍA EN SISTEMAS

IS912 - Desarrollo Web
Ejercicios Prácticos

Catedrático: Erick Vladimir Marín

Ejercicio 6

| | |
|-----------------------|--|
| Proyecto base | |
| Lenguajes/Tecnologías | NodeJS, HTML, CSS, JS, AJAX, JSON, MySQL |

Se le provee los siguientes insumos:

- Proyecto base (Plantilla y html), en este proyecto ya están instalados los modulos express, mysql y body-parser.
- Base de datos (Modelo y export en sql)

Importe la base de datos en mysql. Desde phpmyadmin cree una base de datos vacía con el nombre **db_twitter**, luego seleccione la base de datos creada y haga click en el menú **import**, seleccione el archivo llamado db_twitter.sql y luego en Go.

Copie la carpeta llamada twitter en la carpeta del servidor apache (www), luego acceda a la dirección <http://localhost/twitter> para visualizar el proyecto base proveído.

El objetivo del ejercicio es simular el funcionamiento de la red social twitter. Acceda a <http://localhost/twitter>, el resultado deberá ser similar a la siguiente imagen:

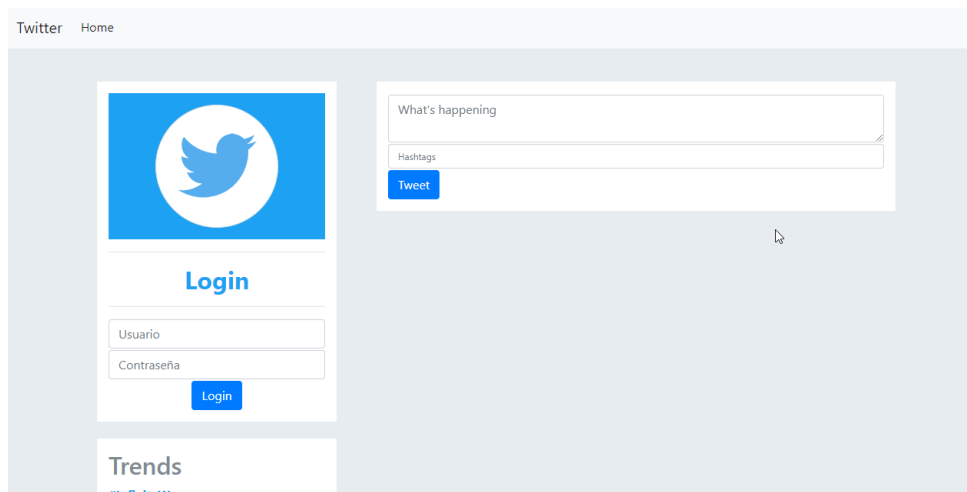


Imagen 1: Pantalla antes de hacer login

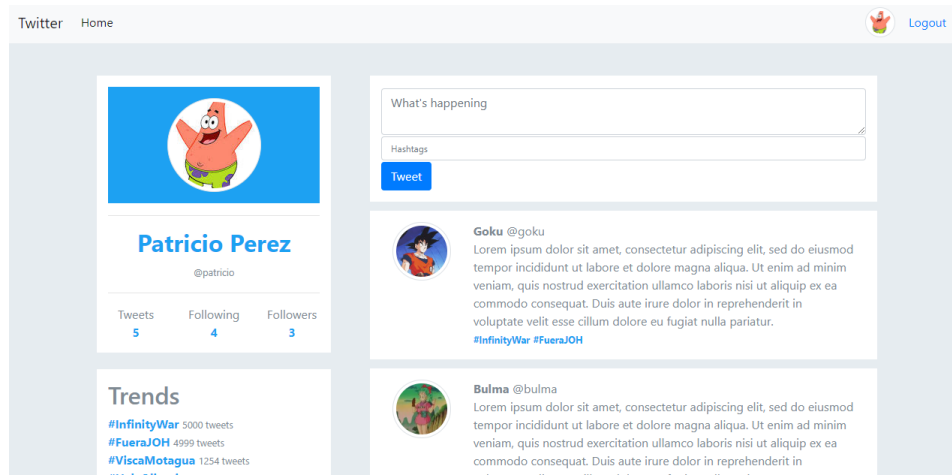


Imagen 2: Pantalla antes de hacer login

Actualmente tienen funcionalidad los botones login y logout que únicamente muestran y ocultan ciertas regiones.

Agregue las siguientes funcionalidades, para cada inciso utilizar peticiones Ajax y respuestas en formato JSON, utilizar conceptos de clases encapsulando las funcionalidades correspondientes:

1. Agregar funcionalidad al botón login para que verifique si el usuario existe en la tabla `tbl_usuarios`, en caso de existir que retorne toda la información del usuario y la ubique en los componentes respectivos del html (Nombre, Nickname, Tweets). Puede guardar el código del usuario en algún input oculto para identificar el código del usuario logueado y este sirva para las siguientes funcionalidades.
2. Una vez logueado el usuario ejecutar una petición ajax para obtener los tweets asociados. Haga dicha petición inmediatamente el usuario haga login. Para obtener los tweets que puede visualizar el usuario actual utilice la siguiente consulta sql (Parametrice los valores que dicen `#USUARIO SELECCIONADO#`):

```
SELECT a.codigo_tweet, b.nombre, b.nickname,
       a.codigo_usuario, a.contenido, a.hashtags, a.fecha
FROM tbl_tweets a
INNER JOIN tbl_usuarios b
ON (a.codigo_usuario = b.codigo_usuario)
WHERE a.codigo_usuario = #USUARIO SELECCIONADO#
OR a.codigo_usuario in (
  select codigo_usuario_sigue from tbl_seguidores
  where codigo_usuario = #USUARIO SELECCIONADO#
);
```

3. En la sección con el título **Trends** generar el contenido al cargar la página utilizando la tabla `tbl_hashtags_trends`.
4. Agregar funcionalidad al formulario para agregar tweets.
5. Al momento de **visualizar** un tweet se debe verificar si en la cadena existen ciertos conjuntos de caracteres o shortcuts que hacen referencia a imágenes llamadas emojis, dependiendo del shortcut escrito el texto debe ser sustituido por la etiqueta html correspondiente para mostrar una imagen. Puede utilizar la función javascript `replace`.

(https://www.w3schools.com/jsref/jsref_replace.asp)

Las sustituciones a realizar son las siguientes:

| | | |
|----------|-------|-------------|
| Shortcut | Emoji | Ruta imagen |
|----------|-------|-------------|










UNAH

UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE
INGENIERÍA EN SISTEMAS

IS912 - Desarrollo Web
Ejercicios Prácticos

Catedrático: Erick Vladimir Marín

| | | |
|-----|---|-----------------------|
| :) |  | img/emojis/emoji1.png |
| XD |  | img/emojis/emoji2.png |
| :P |  | img/emojis/emoji3.png |
| :(|  | img/emojis/emoji4.png |
| :* |  | img/emojis/emoji5.png |
| X_X |  | img/emojis/emoji6.png |
| ** |  | img/emojis/emoji7.png |

Agregar funcionalidad al botón logout para que limpie todos los componentes para que se pueda hacer de nuevo login.



| | |
|-----------------------|--|
| Ejercicio | 6 |
| Proyecto base | |
| Lenguajes/Tecnologías | NodeJS, HTML, CSS, JS, AJAX, JSON, MySQL |

Se le provee los siguientes insumos:

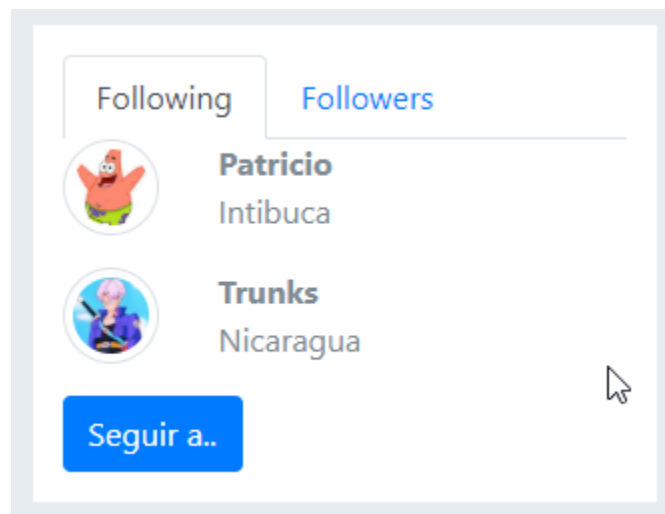
- Proyecto base (Funcional), en este proyecto ya están instalados los modulos express, mysql y body-parser.
- Base de datos (Modelo y export en sql).

El proyecto proveído es funcional, es posible hacer login, mostrar los tweets, trends entre otros utilizando información de la base de datos.

Importe la base de datos en mysql. Desde phpmyadmin cree una base de datos vacía con el nombre **db_twitter**, luego seleccione la base de datos creada y haga click en el menú **import**, seleccione el archivo llamado db_twitter.sql y luego en Go.

Agregue las siguientes funcionalidades, para cada inciso utilizar **peticiones Ajax y respuestas en formato JSON**, utilizar **NodeJS** como plataforma del lado del servidor:

6. En el extremo inferior izquierdo hay un panel en el cual se pueden visualizar las personas que un usuario sigue (Following) y las los usuarios que lo siguen (Followers). Llenar ambas listas con la información de la tabla tbl_seguidores.



7. Agregar la funcionalidad a la ventana modal que aparece al dar click en "Seguir a.." para visualizar la lista de usuario y al dar click en el botón seguir de la ventana modal se agregue un registro en la tabla tbl_seguidores.
8. Dar funcionalidad al botón de editar registro de usuario mostrado en la siguiente figura. Debe editar nombre, apellido, nombre de usuario y lugar.



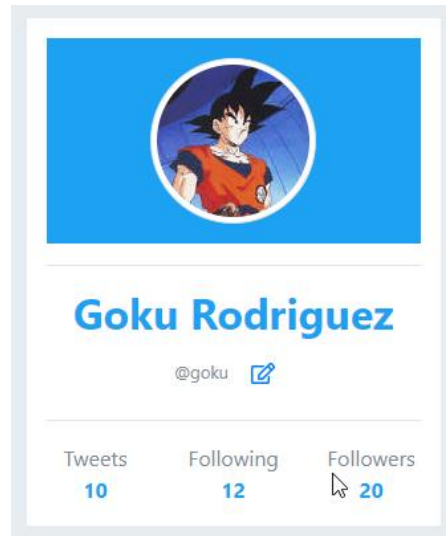
UNAH

UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE
INGENIERÍA EN SISTEMAS

IS912 - Desarrollo Web
Ejercicios Prácticos

Catedrático: Erick Vladimir Marín



9. Agregar funcionalidad al botón like que aparece debajo de cada tweet, esto deberá almacenar un registro en la tabla tbl_likes y actualizar la cantidad de likes del post respectivo.

