

LABORATORIO III - PRIMER PARCIAL

Crear una aplicación de Blog que tenga como mínimo 2 páginas: login.html e index.html.

1) La página de log in recibirá email y password y redireccionará a la página index.html al hacerse click en el botón "Sign in".

- a. Utilizar Bootstrap.
- b. Al hacerse click en el botón "Sign in" se envían los datos en una request AJAX(POST). Los datos deben viajar con el siguiente formato:

```
var datosLogin = {  
  email: email,  
  password: password  
}
```

url servidor login: <http://localhost:1337/login>

2)El servidor retornará valores de preferencias de usuario. Redireccionar a index.html pasándolos como parámetros en la URL (siempre y cuando la propiedad "autenticado" del objeto que retorna el servidor tenga el valor "si"). También se deberá enviar como parámetro el email del usuario.

NOTA: para redireccionar la navegación a index.html:

```
window.location.replace(url);
```

3)Crear la página index.html. Esta página recibirá los parámetros de preferencias de usuario en la URL, así como el email del usuario logeado. Esta página tendrá los controles necesarios para poder realizar un nuevo post en el BLOG.

- a. Esta página deberá contar con un diseño "responsive"(usar sistema de grillas de Bootstrap).
- b. Para obtener los valores de los parámetros que vienen en la URL usar la siguiente función:

```
function getParameterByName(name, url) {  
  if (!url) url = window.location.href;  
  name = name.replace(/[\\[\]]/g, "\\$&");  
  var regex = new RegExp("[?&]" + name + "([^\&#]*)|&|#|$)",  
    results = regex.exec(url);  
  if (!results) return null;  
  if (!results[2]) return '';  
  return decodeURIComponent(results[2].replace(/\+/g, " "));  
}
```

- c. Al momento de submitir el nuevo post, se enviarán al servidor (sin bloquear el navegador) los siguientes valores:

```
var datosPost = {  
  "title": texttitle,  
  "header": textheader,  
  "posttext": posttext,  
  "author": author  
}
```


url servidor para postear una nueva entrada en el blog:

<http://localhost:1337/postearNuevaEntrada>

- d. El servidor devolverá el mismo objeto, agregándole un atributo llamado “date” que contendrá la fecha y hora del servidor. Esta información servirá para agregar al nuevo posteo.
- e. Mientras el servidor procesa la petición, mostrar en pantalla un spinner.
- f. Cuando el servidor retorne los valores, construir el html necesario para mostrar el post en pantalla. Esto implicará sacar el spinner.

LOGIN.HTML

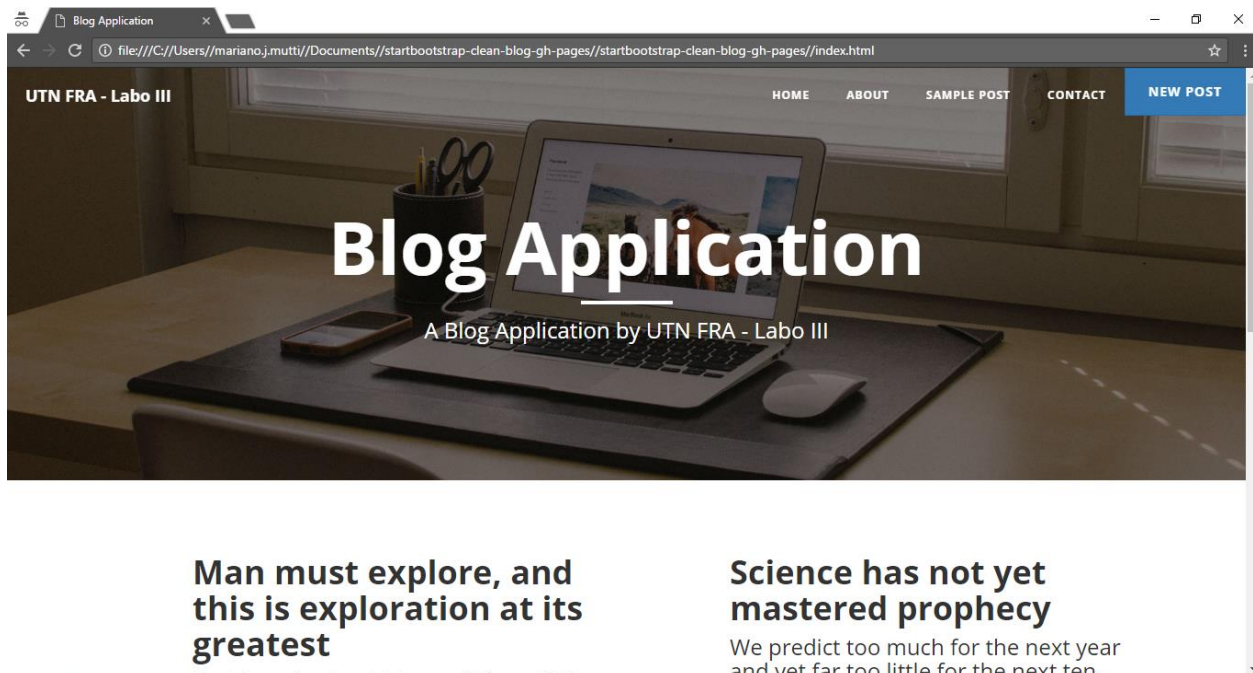
Sign in to continue to main.html

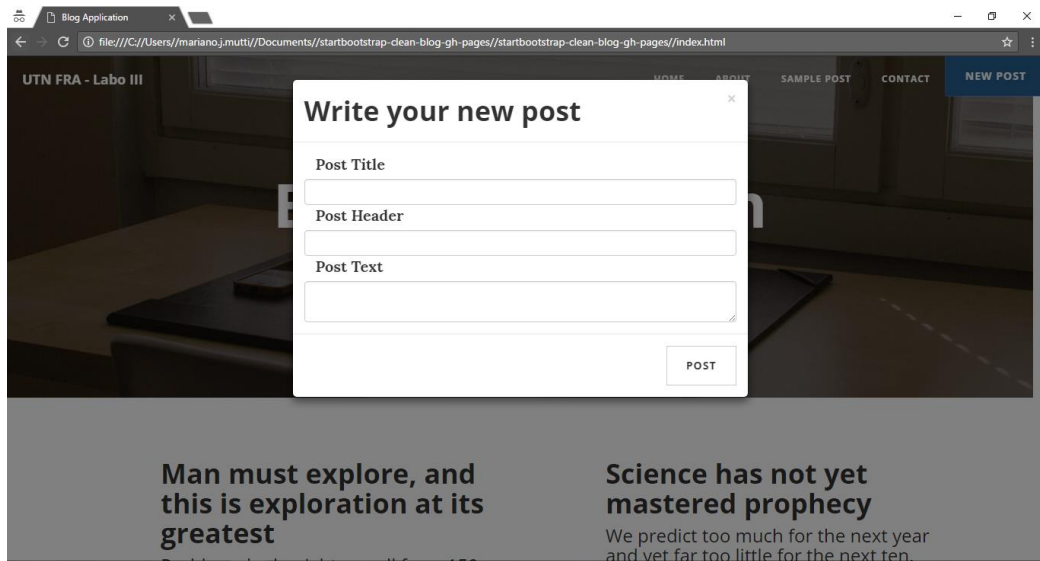


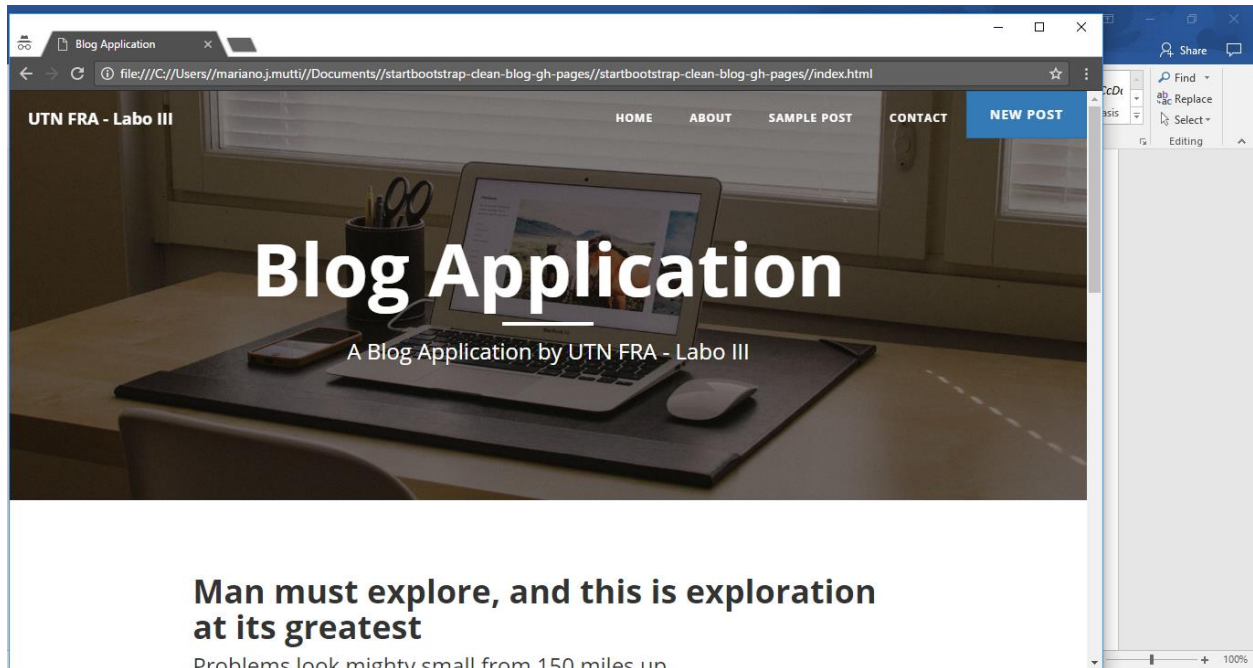
☐ Remember me [Need help?](#)

[Create an account](#)

INDEX.HTML







NOTA: PARA CORRER EL SERVIDOR:

- a. EJECUTAR LA CONSOLA DE NODE JS
- b. NAVEGAR HACIA EL DIRECTORIO DONDE SE ENCUENTRA EL ARCHIVO SERVER.JS
- c. CORRER EL COMANDO "NODE SERVER.JS"

EJEMPLO:

```
Node.js command prompt
Your environment has been set up for using Node.js 4.2.3 (ia32) and npm.
C:\Users\mariano.j.mutti>node server.js
```