

**Ingeniería en Sistemas**

**Computacionales**

**Raúl Tostado Blázquez I.D. 145901**

**Materia: Sistemas Distribuidos**

**Nombre del Trabajo: Práctica Meteor**

**Profesora: Ofelia D. Cervantes Villagómez**

**San Andrés Cholula, Puebla a**

**12 de mayo de 2015**

Reporte práctica Meteor

**Introducción a la práctica**

**Meteor** es una plataforma muy completa para desarrollar aplicaciones móviles y Web únicamente utilizando JavaScript. Algunas características de **Meteor**:

* **Interfaz de usuario moderna**
  + Construye aplicaciones de alta calidad, como Facebook ó Twitter.
* **Mucho menos código**
  + Gracias a su modelo reactivo, desarrolla en pocas líneas lo que tomaría mucho más tiempo.
* **Un lenguaje en todas partes**
  + El mismo lenguaje se utiliza para el **cliente** como para el **servidor**.
* **Actualizaciones en tiempo real**
  + Los usuarios pueden observar cómo cambian los datos instantáneamente y colaborar sin interrupciones.
* **Altamente responsivo**
  + El código en JavaScript se adapta a cualquier plataforma.

En esta actividad desarrollarás una pequeña lista reactiva de "cosas por hacer", en **Meteor** y**MongoDB**.

**Antes de comenzar:**

El primer paso fue dar una lectura detallada a la práctica para saber cuál es el contenido de la misma, posteriormente se descargaron como Zip los programas con los que se trabajó y otros complementos necesarios para trabajar.

**Actividad 1 - Tutorial**

En **Meteor** es muy fácil crear un proyecto nuevo, basta con ir a la terminal y escribir:

meteor create simple-todos

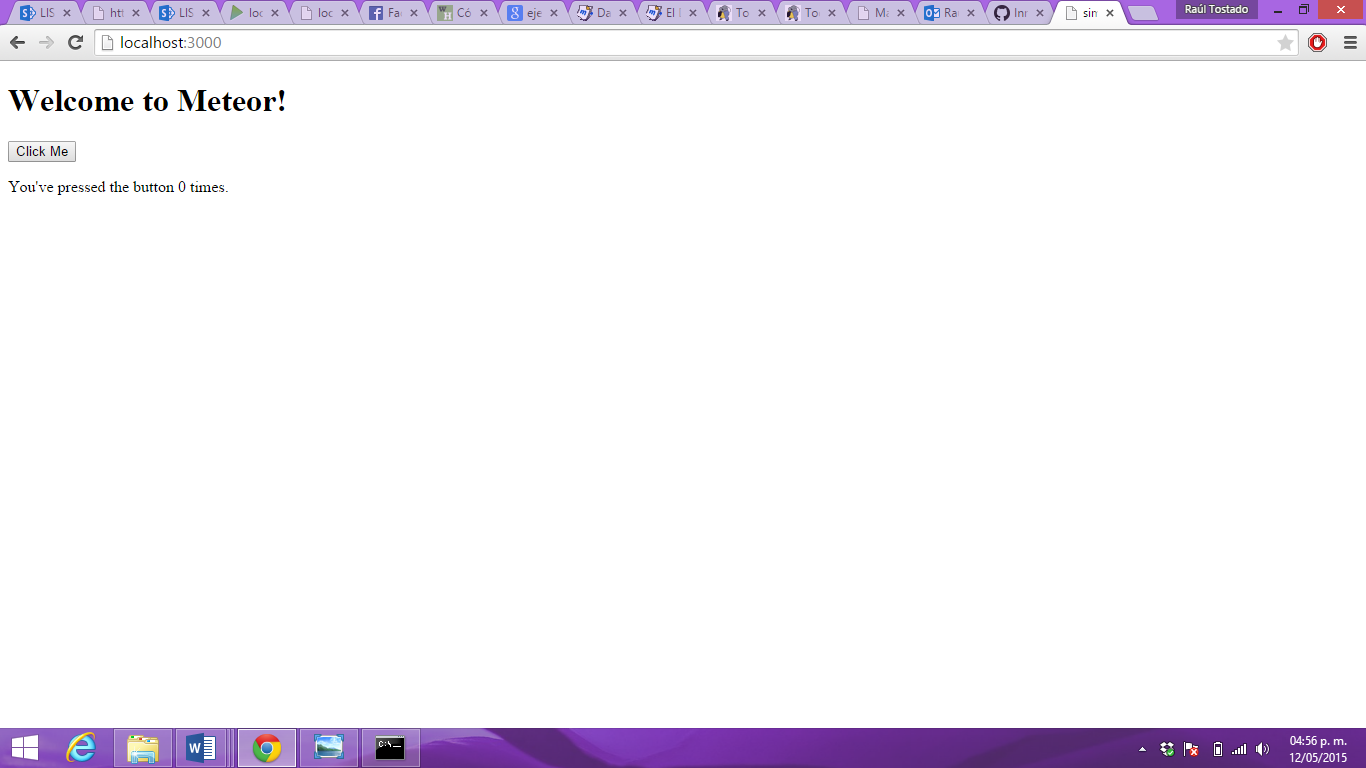
Esto creará una nueva carpeta llamada simple-todos con todos los archivos que la aplicación de **Meteor** necesita para una básica ejecución:

Para ejecutar la aplicación simplemente escribe en la terminal:

cd simple-todos

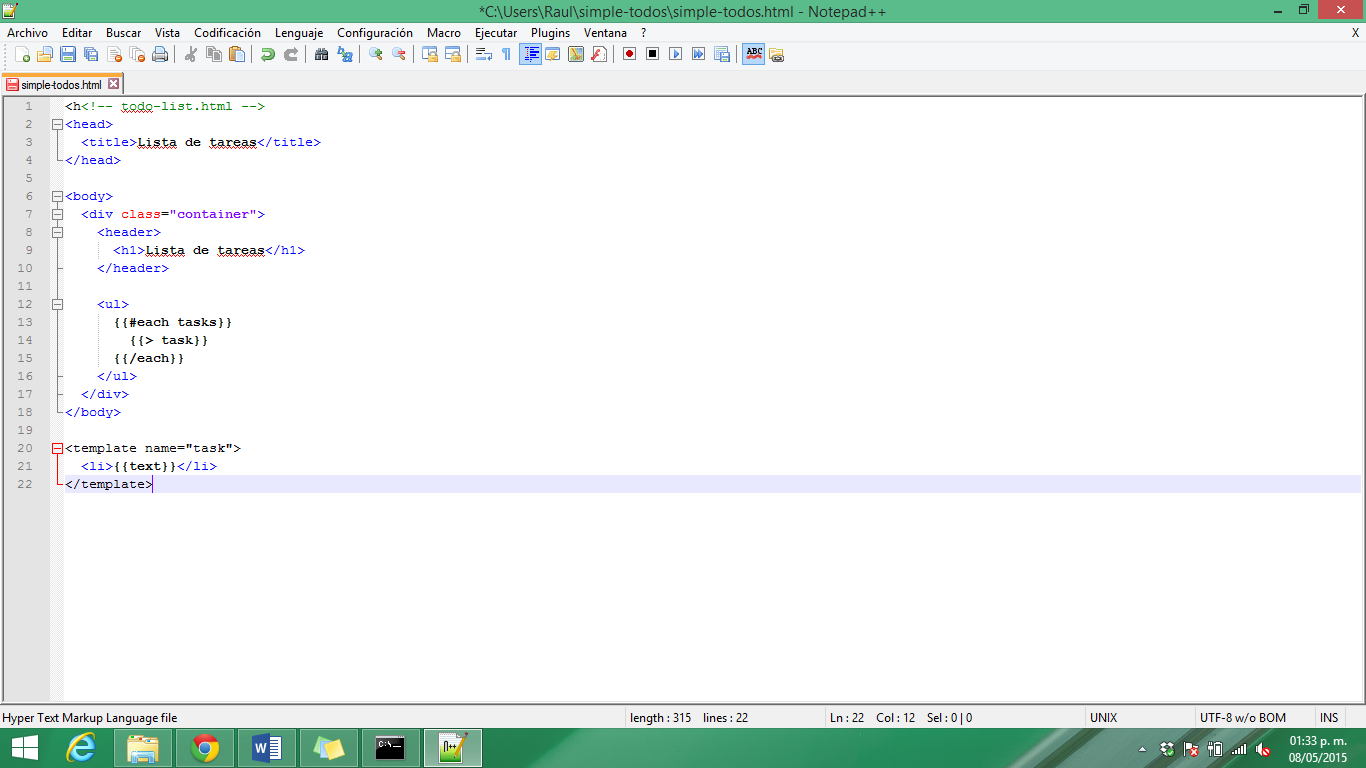
meteor

Abre el navegador de tu preferencia y ve a la siguiente dirección: [**http://localhost:3000**](http://localhost:3000)



**Creando vistas como templates:**

Para comenzar a trabajar con la lista de tareas ahora deberás cambiar el código generado automáticamente por el siguiente:



**Meteor** define vistas a través de **templates**, como puedes observar el template reproduce cada línea para cada tarea.

// todo-list.js

if (Meteor.isClient) {

// Este código se ejecuta únicamente en el cliente

Template.body.helpers({

tasks: [

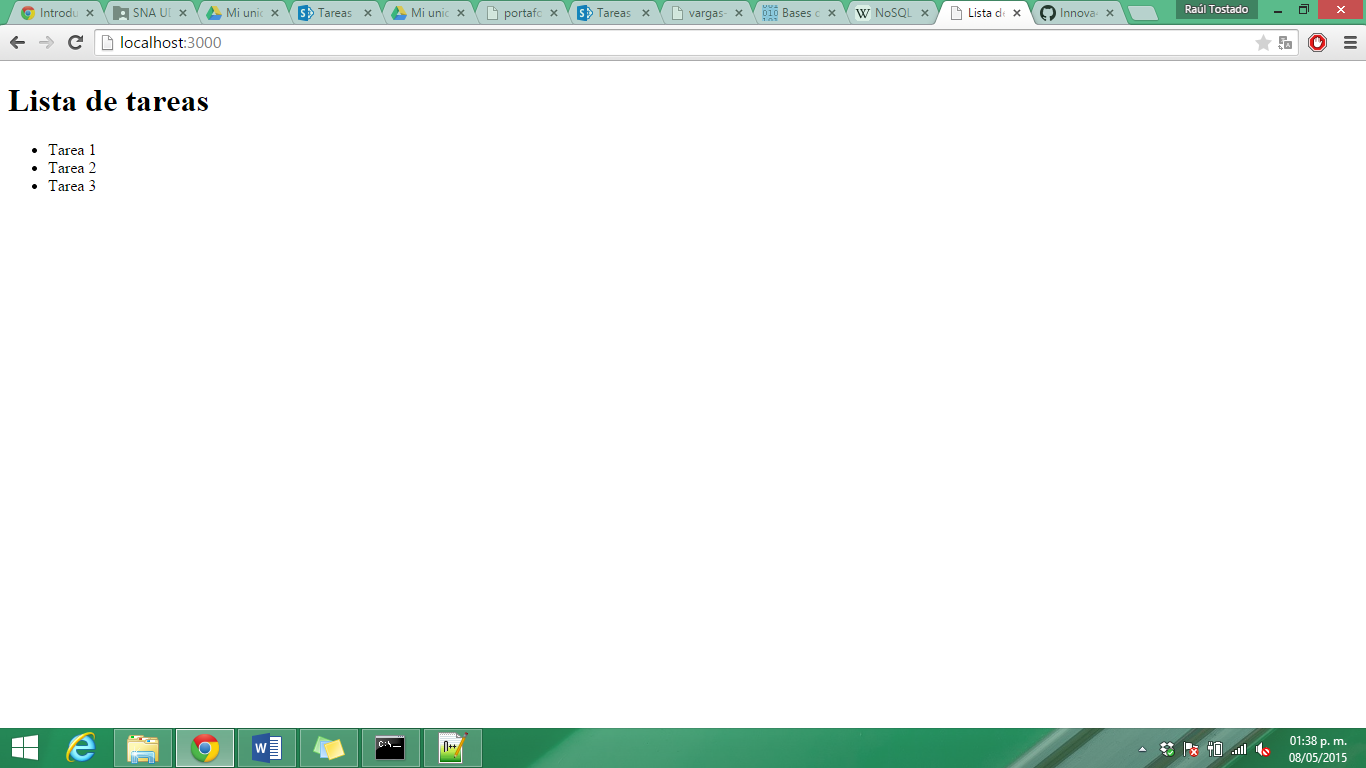
{ text: "Tarea 1" },

{ text: "Tarea 2" },

{ text: "Tarea 3" }

]

}); }



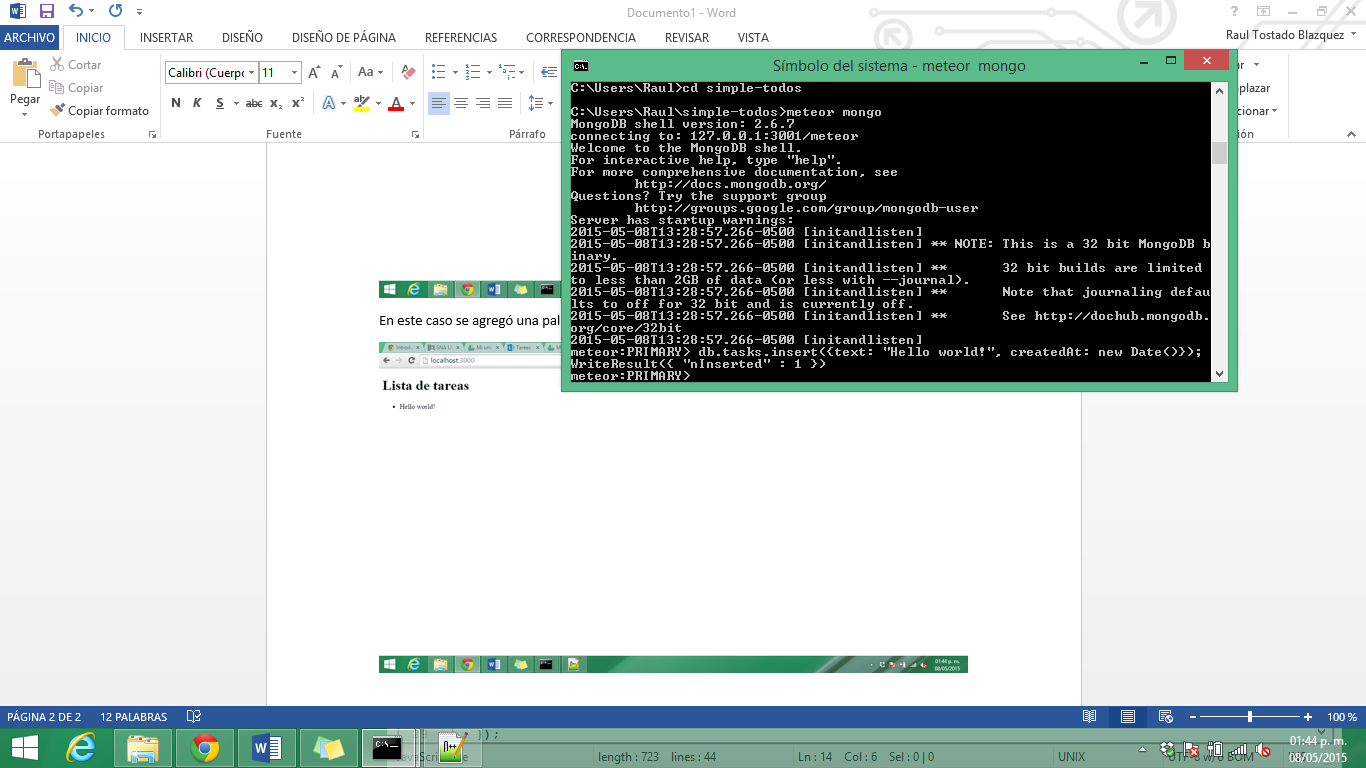
El código **JavaScript** se ejecuta en el **cliente**, por ahora únicamente agregaremos manualmente un par de tareas.

**Conectando con la base de datos**

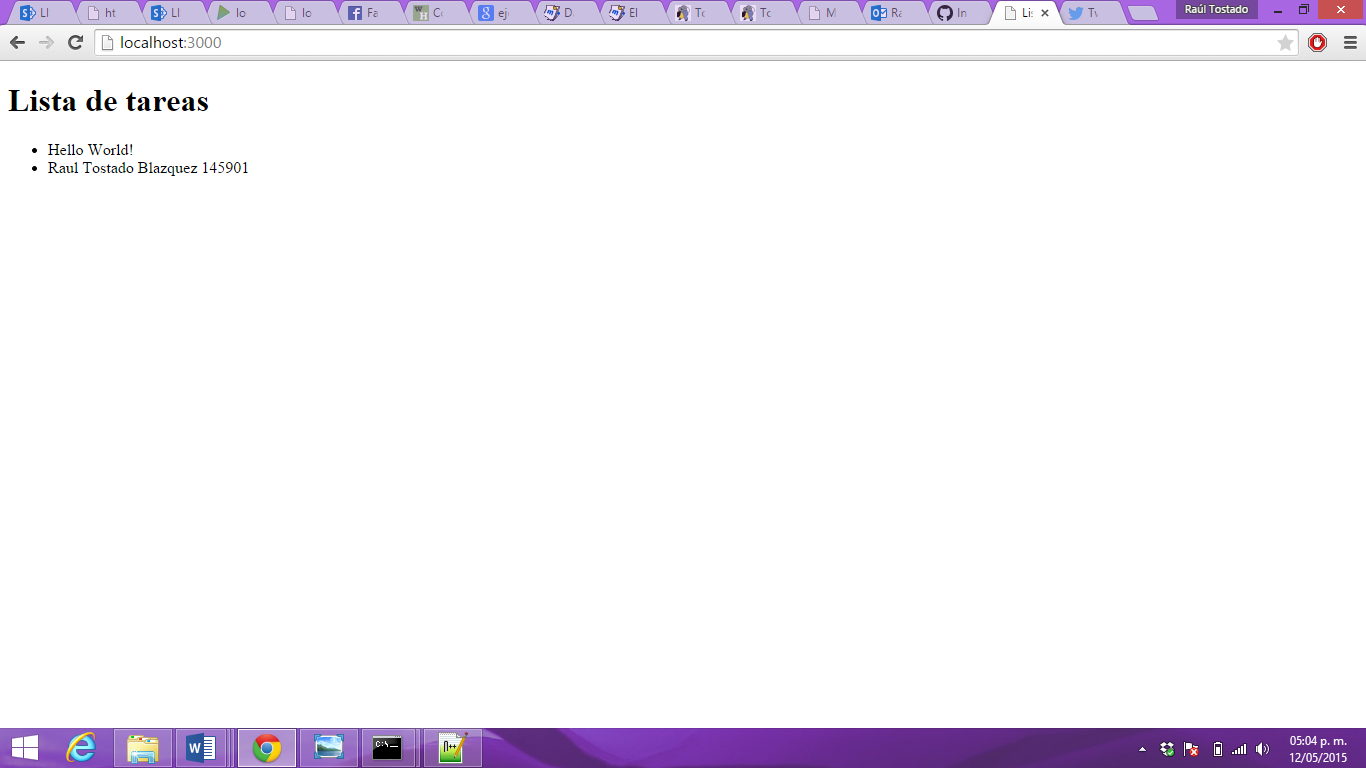
**Meteor** utiliza colecciones de **MongoDB** para almacenar un conjunto de datos. **MongoDB** un sistema de base de datos NOSQL orientado a documentos. **MongoDB** guarda estructuras de datos en documentos tipo **JSON** con un esquema dinámico, por lo que es muy amigable con aplicaciones **JavaScript**.

Ahora modifiquemos el código JavaScript para crear una colección en MongoDB y agregar tareas a nuestra lista, por ahora esto no funcionará, debemos trabajar con la base de datos.

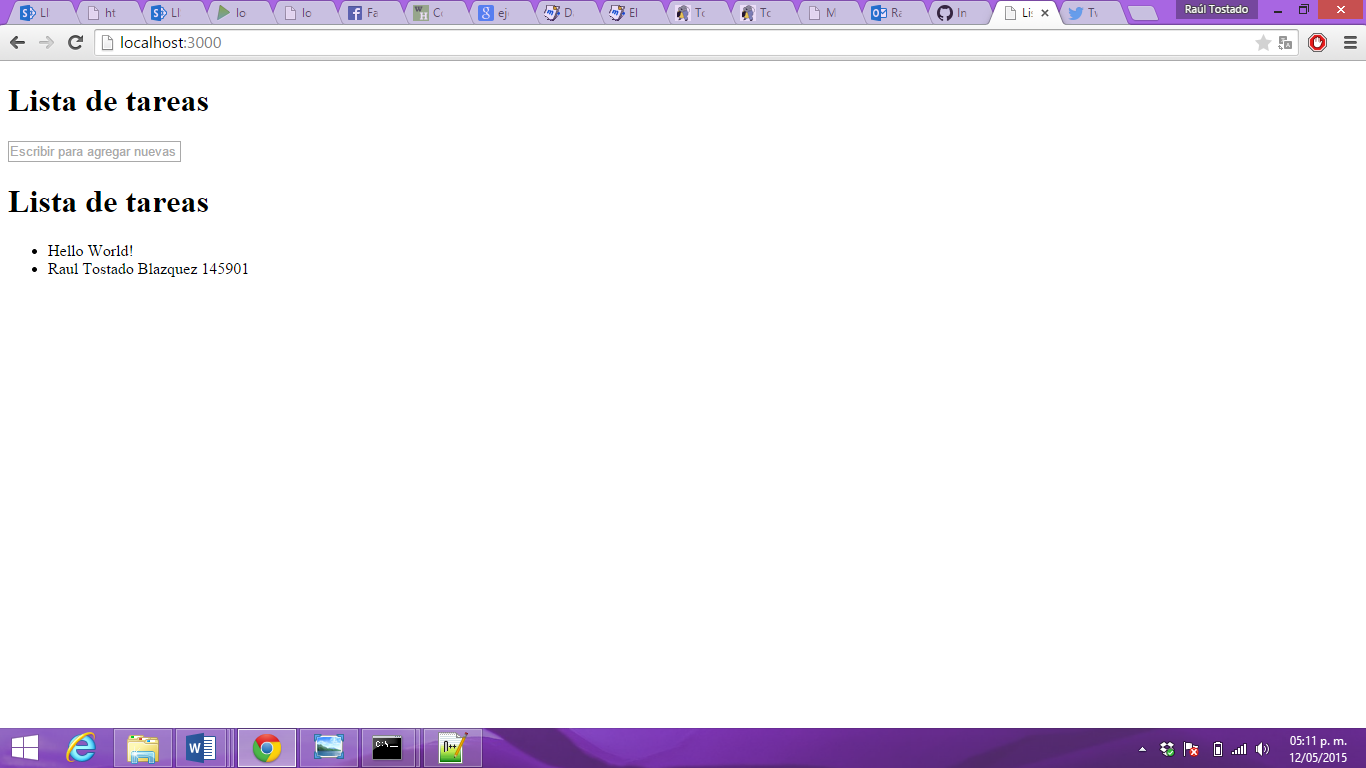
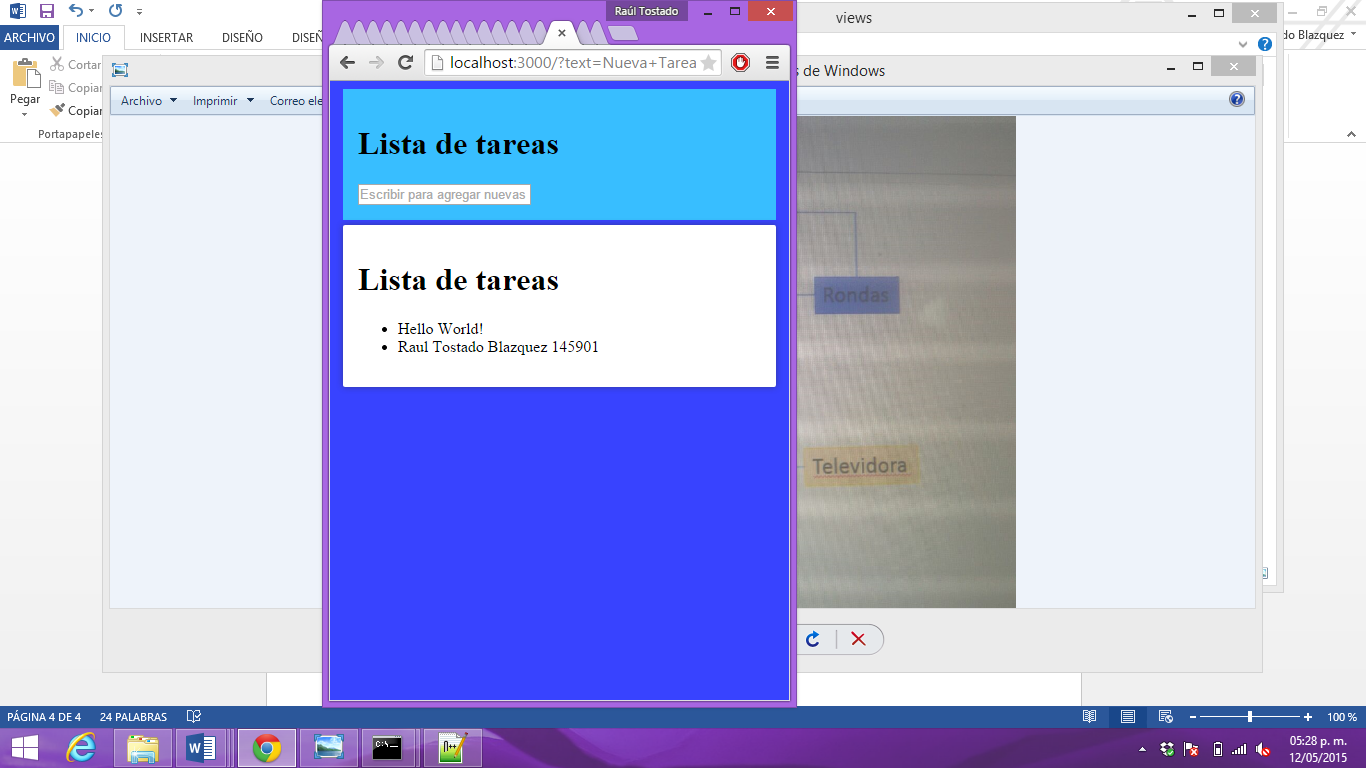
En **Mongo DB** los elementos dentro de una colección se llaman documentos. Abre una nueva terminal para comenzar a utilizar la base de datos y crear algunos documentos que podremos visualizar en la lista.



Abre tu navegador en **localhost:3000** y podrás observar como se ha creado una nueva entrada en la lista de manera automática.



Listo, la aplicación con **Meteor** está casi terminada, ahora deberás editar el código CSS para que la aplicación se vea de la siguiente manera:

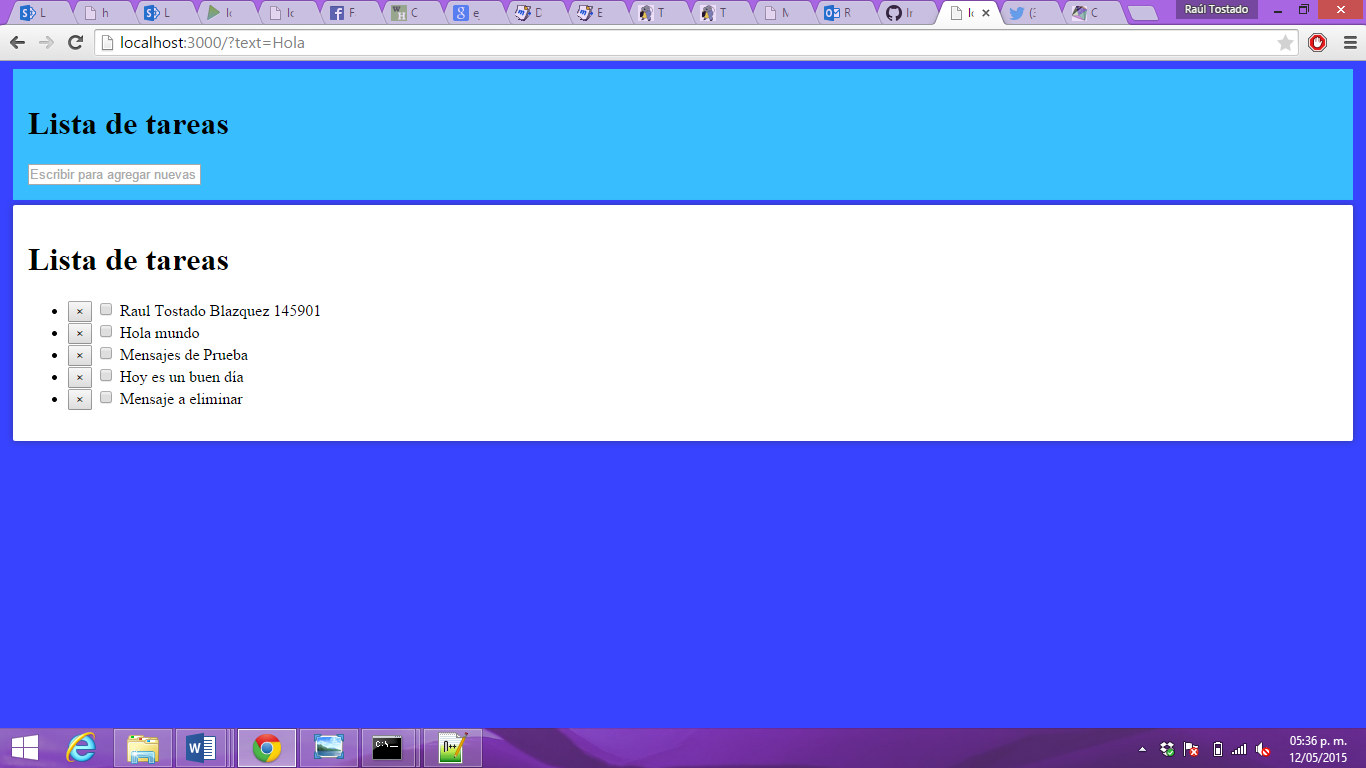
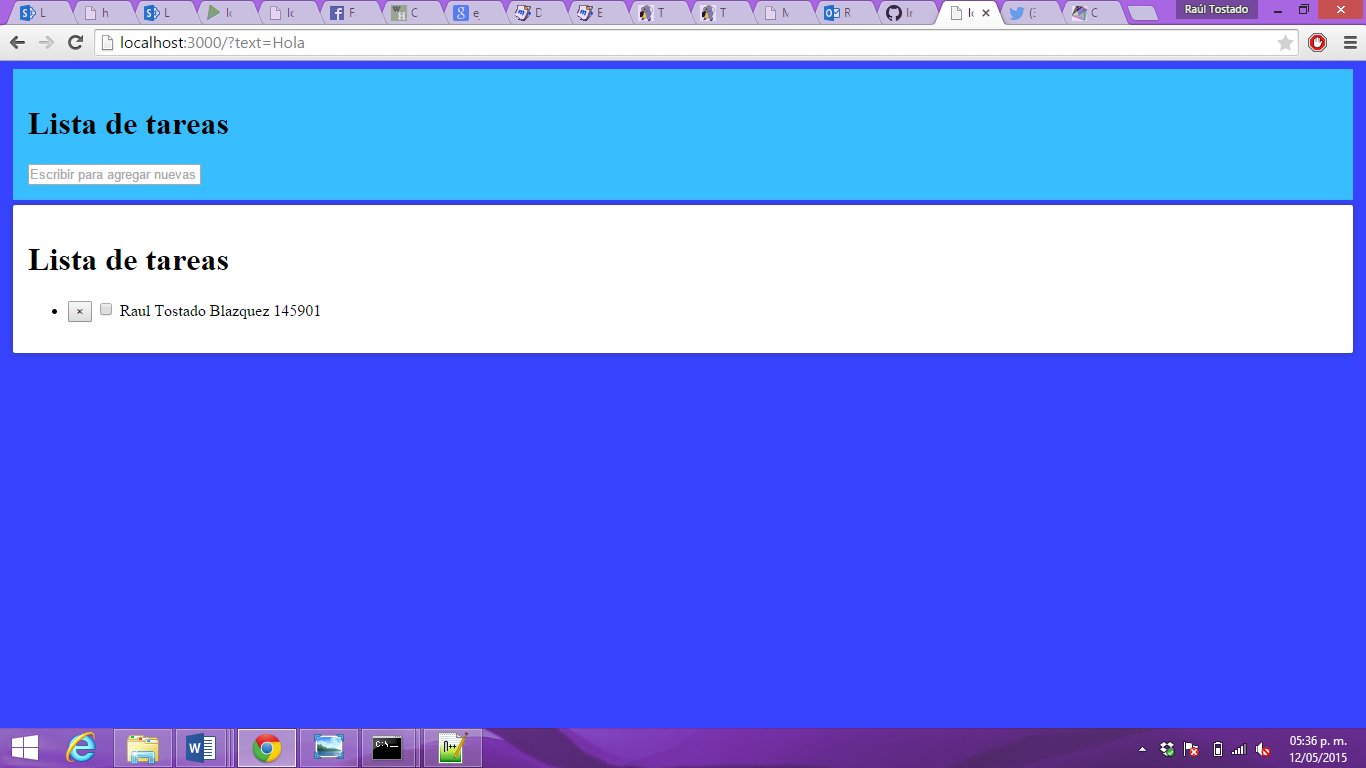
(Antes y después de la hoja de estilo, se intentó igualar a la presentada, pero por no saber exactamente cuáles son los colores, se utilizaron algunos similares).

**Seleccionando y quitando tareas**

Hasta ahora nuestra aplicación agrega tareas, ahora vamos a aprender a eliminarlas, para esto en el template deberás agregar algunos nuevos elementos:

El código ahora es visible en la interfaz pero no hace nada, para esto deberás agregar el comportamiento correspondiente dentro del archivo javascript.

Ahora tienes una **aplicación Web** completa, la aplicación terminada debería verse de la siguiente manera:

**Actividad 2 - Preguntas**

Responder ampliamente a las siguientes preguntas en el reporte, tomando en cuenta todas las actividades realizadas en este repositorio:

* ¿Qué es una aplicación reactiva?

**R=** Una aplicación reactiva es aquella en la que si el programador hace un cambio, es visible para todos los usuarios, también si se añade un nuevo registro, lo pueden visualizar todos los usuarios, un ejemplo es un chat que al escribir algo, es visible para todos.

* ¿Qué es **MongoDB**?

**R=** **MongoDB es**  un sistema de base de datos NoSQL orientado a documentos. **MongoDB**guarda estructuras de datos en documentos tipo **JSON**con un esquema dinámico, por lo que es muy amigable con aplicaciones **JavaScript.**

* Explica el uso de los templates.

R= Gracias a los templates vamos a generar las vistas que usa Meteor y así llamar a las funciones con las que trabajaremos

* Describe ampliamente la conexión "automática" entre **Meteor** y **MongoDB**.

R= Dentro de la carpeta que se trabaja la aplicación, se debe iniciar la base de datos de MongoDB, cabe resaltar que se usa en este caso una ventana de comandos diferente, ya en la aplicación el comando new Mongo.Collection("tasks") para que con esto estén sincronizados y podamos añadir uno a uno los elementos de esta colección

* ¿Qué tendrías que hacer para desarrollar la aplicación en **Play Framework**?

R= Una idea sería crear el servicio y posteriormente llamarlo indicando cual es el texto que se va a mandar para que aparezca en la nueva conexión

* Explica cómo se realiza el registro de nuevas tareas en **MongoDB**.

R= Es un comando un poco largo, declaras la colección en la que se va a guardar, el texto que tendrá, la fecha como AT y se cierran los signos necesarios para que se pueda trabajar, se puede agregar ya sea desde la aplicación o desde la consola.

* Explica cómo se realiza la eliminación de tareas en **Meteor** con **MongoDB**.

R= Hay una casilla de verificación, en cuanto se presiona llama a la acción de eliminar dentro de la aplicación

* ¿En qué otras aplicaciones podrías utilizar **Meteor**?

R= En alguna aplicación móvil que requiera el constante mantenimiento de la misma, así como la actualización necesaria para los usuarios.

**Conclusiones**

Meteor tiene una similitud con las prácticas anteriores, pero cuenta con la ventaja que no es necesario actualizar el navegador en el momento en que hay un cambio, se modifica instantáneamente; tiene sus ventajas y desventajas, pero si aprendemos a manejarla de manera óptima, lograremos sacarle el mayor provecho a la misma