|  |  |  |  |
| --- | --- | --- | --- |
| logo_udlap    **EDEI** | **Reporte Meteor** | | |
| **DEPARTAMENTO** | Computación, Electrónica y Mecatrónica | **MATERIA** | Sistemas Distribuidos |
| **PROFESOR** | Dra. Ofelia Cervantes | **PERIODO** | Primavera 2015 |

Introducción

**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.

Objetivo

Es crear una lista reactiva en esta lista tendremos tareas pendientes que no hemos completado.

Desarrollo

Primero debemos de instalar Meteor en nuestro equipo, el instalador lo obtendremos de la siguiente dirección:

<https://install.meteor.com/windows>

Una vez descargado debemos de ejecutar el instalador, cuando el procesos termine las computadoras que tengan Windows 8.1 deberán ser reiniciadas para que los comandos funcionen correctamente.

Cuando la computadora este lista debemos de colocarnos en un nuestra carpeta de preferencia, atreves de una ventana de comandos debemos de llegar a ese directorio y ejecutar el siguiente comando:

meteor create simple-todos

En esta ocasión el nombre de nuestra aplicación será simple-todos, cuando la ejecución termine, se generara los siguientes archivos.

//////foto

Para ejecutar la aplicación debemos de entrar al directorio que creo el primer comando y ejecutar el siguiente tal como lo muestra el siguiente bloque de instrucciones.

cd simple-todos

meteor

En nuestro navegador web de nuestra preferencia debemos de ir a la siguiente dirección para ver el resultado de nuestra aplicación.

**http://localhost:3000**

Para continuar debemos de insertar el siguiente código en el archivo todo-list.html y todo-list.js y simple-todos.js.

<!-- todo-list.html -->

<head>

<title>Lista de tareas</title>

</head>

<body>

<div class="container">

<header>

<h1>Lista de tareas</h1>

</header>

<ul>

{{#each tasks}}

{{> task}}

{{/each}}

</ul>

</div>

</body>

<template name="task">

<li>{{text}}</li>

</template>

// 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" }

]

});

}

// simple-todos.js

Tasks = new Mongo.Collection("tasks");

if (Meteor.isClient) {

// Este código se ejecuta únicamente en el cliente

Template.body.helpers({

tasks: function () {

return Tasks.find({});

}

});

}

Continuamos abriendo una nueva ventana de comandos en la que ejecutaremos los siguientes comandos para crear nuevos archivos en la base de datos.

meteor mongo

db.tasks.insert({ text: "Hello world!", createdAt: new Date() });

Debemos de completar el siguiente código en los archivos correspondientes.

<header>

<h1>Lista de tareas</h1>

<!-- Agregar un formulario después de H1 -->

<form class="new-task">

<input type="text" name="text" placeholder="Escribir para agregar nuevas tareas" />

</form>

</header>

// Dentro de el bloque if (Meteor.isClient) después de Template.body.helpers:

Template.body.events({

"submit .new-task": function (event) {

// Esta función será llamada despues de agregar un nuevo elemento

var text = event.target.text.value;

Tasks.insert({

text: text,

createdAt: new Date() // current time

});

// Clear form

event.target.text.value = "";

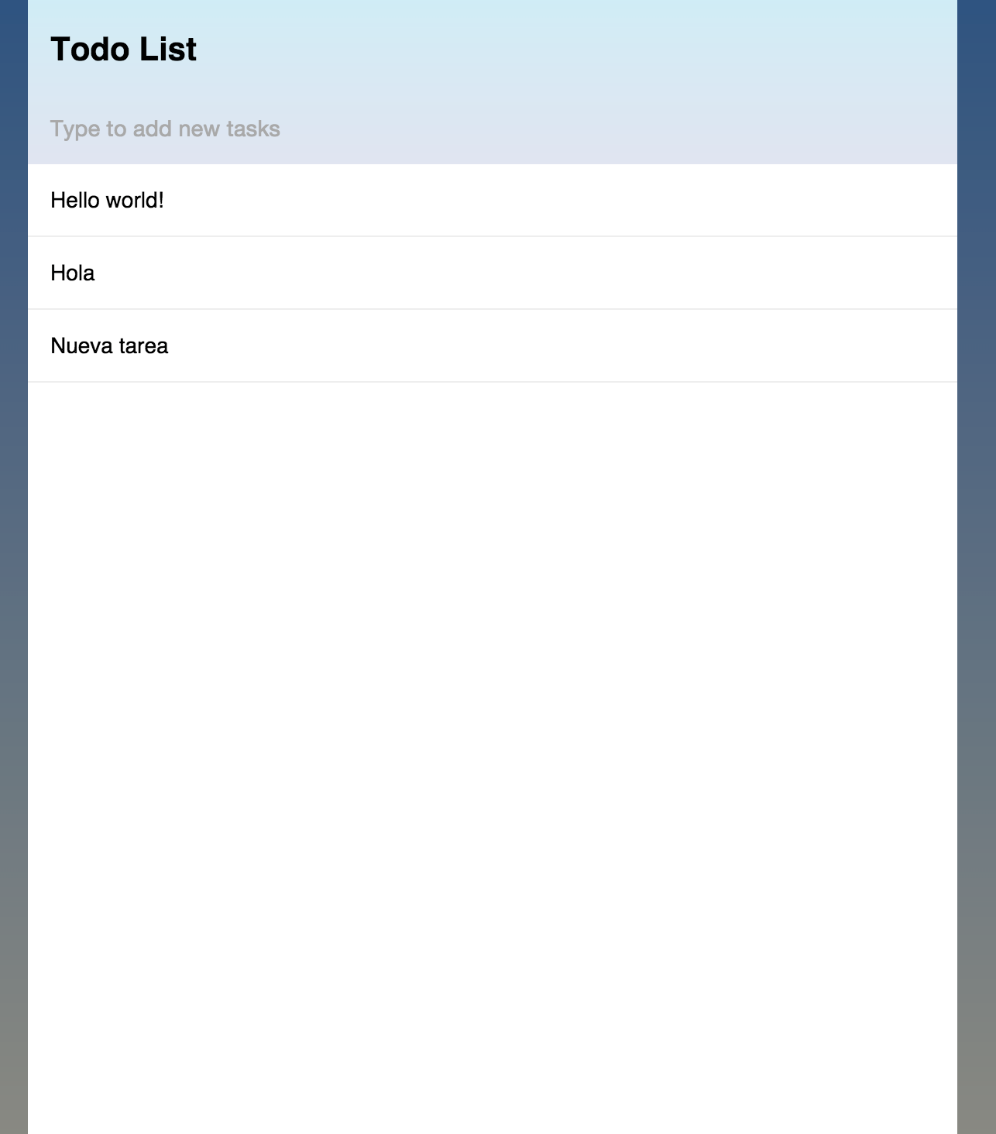
// Prevenir que se envíe por default

return false;

}

});

El resultado debe verse como la siguiente imagen.



Para poder eliminar tareas en la lista, debemos de insertar el siguiente código.

<!-- reemplazar el template actual por el siguiente -->

<template name="task">

<li class="{{#if checked}}checked{{/if}}">

<button class="delete">&times;</button>

<input type="checkbox" checked="{{checked}}" class="toggle-checked" />

<span class="text">{{text}}</span>

</li>

</template>

// En el código del cliente, debajo de todo lo demás.

Template.task.events({

"click .toggle-checked": function () {

Tasks.update(this.\_id, {$set: {checked: ! this.checked}});

},

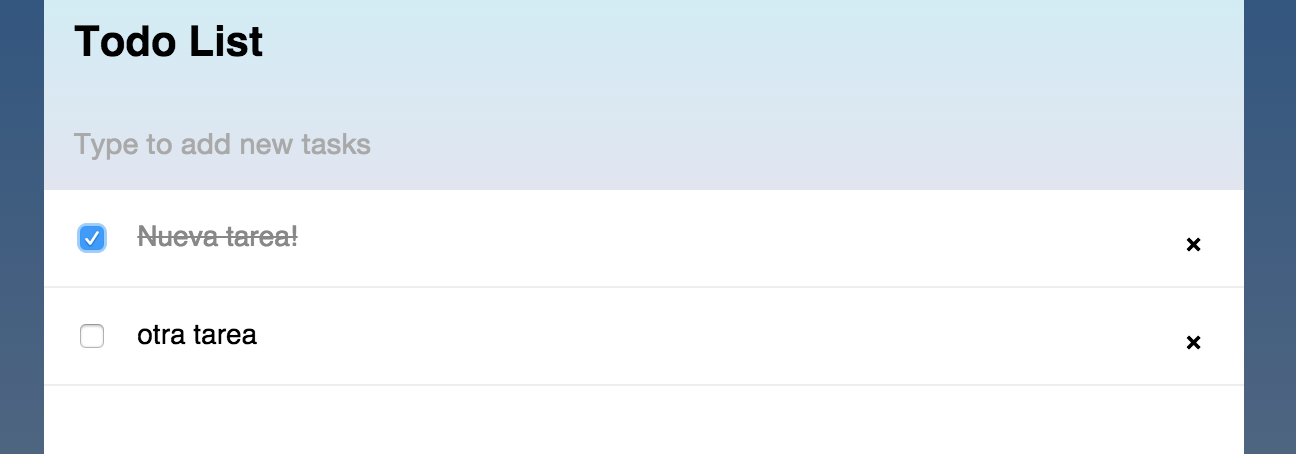
"click .delete": function () {

Tasks.remove(this.\_id);

}

});

La aplicación lista debería de verse de la siguiente manera



Conclusión.

En esta práctica se aprendió a utilizar la tecnología Meteor que nos permite crear aplicación móvil de una manera más fácil, los resultados son de una mejor calidad.

Meteor permite el uso de Java Scrip lo que permite una flexibilidad y que nuestra aplicación se adapte a diferentes plataformas.

Preguntas

* ¿Qué es una aplicación reactiva?
  + Las aplicaciones reactivas son aquellas que tienen componentes activos y listos para recibir eventos, una aplicación reactiva debe ser escalable, sensible, resistente.
* ¿Qué es **MongoDB**?
  + n 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.
  + Un templete permite al cliente pedir información a la base de datos de MongoDB
* Describe ampliamente la conexión "automática" entre **Meteor** y **MongoDB**.
  + Meteor y MongoDB son tecnologías que utilizan como lenguaje javaScript por lo que las tecnologías se acoplan de muy buena forma permitiendo un funcionamiento adecuado y eficiente para el programador
* ¿Qué tendrías que hacer para desarrollar la aplicación en **Play Framework**?
  + Debería de crear un servicio web que recupere los mensajes de una base de datos distribuida y los coloque como tareas, para esto se necesita que otro servicio que utiliza base de datos interactuare con la aplicación, además de que la pagina web debería permitir agregar y eliminar tareas, lo que podría resultar complicado de realizar.