

---

# Table of Contents

Prologo	1.1
El desarrollador excepcional	1.2
¿Por qué Meteor?	1.3
Principios de Meteor	1.4
Programación reactiva	1.5
Transferencia de datos	1.6
Paquetes	1.7
Seguridad	1.8
Despliegue	1.9
Bibliografia	1.10

# Prologo

Este libro sirve como documentación acerca de [Meteor](#), una plataforma para la creación de aplicaciones tanto web como móvil como de escritorio utilizando como lenguaje de programación Javascript.

Se explicará tanto la filosofía de Meteor como las diferentes partes que lo componen.

Tengo que agradecer a toda la gente que ha trabajado para que hoy tengamos Meteor, así como toda la gente que le ha dado su apoyo.

# El desarrollador excepcional

Todo el mundo quiere ser excepcional en su ámbito de trabajo. Según [Larry Wall](#), el creador del lenguaje de programación Perl, todo desarrollador tiene que tener tres cualidades.

1. La pereza: La calidad que te hace esforzarte al máximo para reducir el gasto total de energía. Te hace escribir programas para ahorro de mano de obra que otras personas encontrarán útiles y documentar lo que escribió para no tener que responder a muchas preguntas sobre lo mismo.
2. La impaciencia: La rabia que se siente cuando el ordenador está siendo perezoso. Esto hace escribir programas que no sólo reaccionan a sus necesidades, sino que se anticipan a ellas. O al menos así lo parezca.
3. Orgullo: La calidad que te hace escribir (y mantener) programas que otras personas no sientan la necesidad de decir cosas malas sobre ellos.

Gracias a Meteor podremos potenciar estas cualidades y mejorar como programadores.

## ¿Por qué Meteor?

Año 2016, el desarrollador Full-Stack esta de moda, Javascript esta de moda.

Si hay que elegir un entorno full-stack en Javascript deberíamos buscar diferentes opciones y compararlas. Una comparativa con su competencia directa con fecha 12/06/2016

	Meteor	Derby	Wakanda	Mean.io	Mean.js
Preguntas StackOverflow	21678	100	67	271	718
Estrellas en GitHub	34213	4051	28	9027	3459
Commits en GitHub	16064	1688	0	1460	1199
Forks en GitHub	4219	229	31	2743	1347
Seguidores en Twitter	55217	2103	1418	4232	3121

Recientemente se ha acuñado un termino en la comunidad Javascript. Fatiga de Javascript. Este termino se debe a que para ser un buen programador Javascript hoy en dia se supone que debes conoer todas estas tecnologias.



Y dentro de unos meses aparecerá algo para reemplazar alguna de las piezas y así hasta el fin de los dias. Meteor nos libera de mucha de esta carga.

Un punto fuerte de Meteor es su facilidad de aprendizaje. Meteor ha sido denominado como

Embarazosamente fácil de aprender

A diferencia de Angular caracterizado como

Vendaje sobre heridas auto infligidas



# Principios de Meteor

Meteor se rige por siete principios:

## Los datos sobre el cable

No envíes HTML a través de la red, envía los datos y deja que el cliente decida cómo renderizarlos.

## Un lenguaje: Javascript.

No tener que cambiar de lenguaje cuando se programa tanto el frontend como el backend es una gran ventaja. No hay que estar cambiando el chip mentalmente, siempre se tendrán a tu disposición las mismas funciones en el servidor que en el cliente.

## Bases de datos en todos lados.

Utiliza la misma API para acceder a tu base de datos desde el cliente como desde el servidor.

Así es como definimos una lista de objetos tanto en cliente como en servidor.

```
Lists = new Meteor.Collection("lists");
```

## Compensación de latencia

En el cliente, utilice la obtención previa de datos y la simulación del modelo para que se vea como que tiene una conexión sin latencia con la base de datos.

## Reactividad en todo el sistema

Hacer tiempo real el valor por defecto. Todas las capas, desde la base de datos hasta la presentación, la interfaz debe ser manejada por eventos.

## Abraza el ecosistema.

Meteor es código libre e integra en vez de reemplazar las existentes herramientas de código abierto ya existentes.

## **Simplicidad equivale a productividad**

La mejor manera de hacer parecer algo siempre es que lo sea realmente. Consigue esto a través de APIs limpias y elegantes.

# Programación reactiva

La programación reactiva es un pilar de Meteor. ¿Pero qué es la programación reactiva?

Podemos utilizar como definición:

La programación reactiva es un paradigma de programación que está orientado a los flujos de datos y más específicamente a la propagación de los cambios de los datos en dicho flujo.

Palabra clave propagación.

Aquí un ejemplo en programación no reactiva.

```
var a = 1;
var b = 2;
var c = a + b;

console.log(c); // 3

a = 2;

console.log(c); // 3
```

Ahora el mismo ejemplo en programación reactiva.

```
var a = 1;
var b = 2;
var c = a + b;

console.log(c); // 3

a = 2;

console.log(c); // 4
```

Aquí vemos como al cambiar el valor de a, este cambio se ha propagado a c ya que depende de a.

Meteor pone a nuestra disposición herramientas para utilizar la programación reactiva en todas las partes de nuestra aplicación como son [Tracker](#) y [Reactive-Dict](#)





# Datos en Meteor

El flujo de datos entre cliente y servidor en una aplicación de Meteor se realiza sobre el protocolo DDP.

## DDP

Meteor utiliza un protocolo para comunicar el servidor con el cliente conocido como DDP(Distributed Data Protocol)

Utiliza el patrón de [publicación-suscripción](#)

Es un protocolo muy simple y minimalista, basado en JSON, la capa de transporte que utiliza Meteor es WebSockets y SocktJS, el cual es una emulación de WebSockets cuando no se encuentran disponibles.

Existen muchos clientes que implementan este protocolo para comunicarse con aplicaciones Meteor e igualmente es muy sencillo crear un cliente que lo implemente.

[Aquí](#) tenemos una lista de diferentes clientes DDP

## Base de datos

Meteor viene con MongoDB incorporado, podremos acceder a la base de datos directamente a través de las colecciones de Meteor. En el cliente tendremos una copia de la base de datos a través de MiniMongo lo que algunos consideran una base de datos en el cliente, seria mas apropiado tratar a MiniMongo como un sistema de cache muy sofisticado

Gracias a [Apollo](#) podemos utilizar cualquier base de datos de forma reactiva y no estar limitados a MongoDB

## Publicaciones y Suscripciones

En una aplicación seria en producción deberemos definir que partes de la base de datos queremos exponer en el cliente gracias al sistema de publicaciones y suscripciones. El tiempo real tiene un coste en recursos y para minimizar ese coste y hacer nuestra aplicación eficiente deberemos publicar y suscribirnos solo a la mínima cantidad de datos necesarios.



# Paquetes

Una gran ventaja de Meteor son todos los paquetes que ya tiene escritos lo que nos ahorrara muchísimo tiempo al no tener que escribirlos nosotros. Meteor nos ofrece nuestro propio gestor de paquetes llamado [Atmosphere](#) con una gran cantidad de paquetes muy interesantes. Autenticación, preprocesadores de estilos, routers, validación de formularios... prácticamente cualquier cosa que se nos ocurra a una linea de comando de distancia.

Desde la version 1.3 Meteor tiene soporte oficial a los paquetes de npm, con lo que la cantidad de código que tenemos a nuestra disposición para reutilizar en nuestros proyectos es formidable.

# Seguridad

Tener acceso a la base de datos en el cliente puede dar la impresión de ser poco seguro. Y es cierto. Por defecto Meteor viene con dos paquetes incluidos, [autopublish](#) e [insecure](#). Son dos paquetes que nos facilitaran mucho el tiempo a la hora de hacer un prototipo inicial. Pero que mas adelante tendremos que eliminar y configurar bien la seguridad de nuestra aplicación.

Cuando quitemos estos paquetes ya no podremos realizar ninguna operación sobre la base de datos, ni tendremos acceso a ninguno de los datos. Para tener acceso a los datos tendremos que utilizar las publicaciones y suscripciones

Para realizar acciones sobre la base de datos, ya sea insertar, borrar o actualizar cualquier cosa, tenemos dos opciones. Definir una a una las reglas que permitimos o prohibimos, utilizando metodos de las propias colecciones. O invocar a métodos en el servidor, desde el cliente, y el servidor tiene pleno acceso a la base de datos. La opción de permitir y denegar es muy vulnerable a dejar huecos por los que saltarse la seguridad muy rápidamente, con los métodos tenemos un mayor control para securizar nuestra aplicación.

Existe un paquete muy importante [audit-arguments-check](#) que nos obligara a que todo dato proporcionado por el usuario tenga que ser verificado utilizando la función check que nos ofrece Meteor. Con lo que nos obligamos a cumplir una de las reglas mas importantes en seguridad informática, nunca confíes en lo que te envíe un usuario.

# Despliegue

Meteor nos ofrece la posibilidad de alojar nuestras aplicaciones en sus servidores a través de un servicio conocido como [Galaxy](#). Como al final una aplicación Meteor es una aplicación de Node, siempre podremos desplegarla en todos los sitios en los que estamos acostumbrados a desplegar nuestras aplicaciones Node.

Modulus, Openshif, AWS, OceanDigital, Heroku... Todos ofrecen muchas facilidades a la hora de desplegar nuestra aplicación. Al estar tan en auge Meteor, la comunidad es muy activa y existe mucha información al respecto e incluso herramientas como [meteor-up](#) para hacer el despliegue de aplicaciones realmente fácil.

# Bibliografía

## Enlaces

<https://github.com/Urigo/awesome-meteor>  
<https://github.com/arunoda/meteor-up>  
<https://www.meteor.com/hosting>  
<https://atmospherejs.com/>  
<https://www.npmjs.com/>  
<http://info.meteor.com/blog>  
<https://c.kat.pe/seven-principles-of-meteor-dot-js#.V11xaPI97RZ>  
<http://threevirtues.com/>  
[https://en.wikipedia.org/wiki/Publish%E2%80%93subscribe\\_pattern](https://en.wikipedia.org/wiki/Publish%E2%80%93subscribe_pattern)  
<https://meteorhacks.com/introduction-to-ddp/>  
<http://www.ceiba.com.co/es/articulo-1-del-blog/>  
<http://stackoverflow.com/research/developer-survey-2016>  
[http://www.meteorpedia.com/read/Why\\_Meteor](http://www.meteorpedia.com/read/Why_Meteor)  
<http://guide.meteor.com/index.html>  
<http://www.apollostack.com/>  
<https://meteorhacks.com/introduction-to-ddp/>  
<https://kadira.io/>  
<https://kadirahq.github.io/mantra/>

## Libros

<https://www.packtpub.com/web-development/building-single-page-web-apps-meteor>  
<https://www.discovermeteor.com/>  
<http://meteorandreact.com/book/>

<https://www.packtpub.com/web-development/mastering-meteorjs-application-development>

## **Cursos**

<https://www.udemy.com/learn-meteorjs-by-building-10-real-world-projects/>

<https://www.udemy.com/meteor-ecommerce-site-with-amazon-s3-paypal-integration/>

<https://www.youtube.com/playlist?list=PLLnpHn493BHECNI9I8gwos-hEfFrer7TV>