

Programación Asíncrona en JavaScript

Arquitecturas de Integración

Javier Vélez Reyes

@javiervelezreye
Javier.velez.reyes@gmail.com

Diciembre 2016



Programación Asíncrona en JavaScript

Autor

Programación Asíncrona en JavaScript

Autor

¿Quién Soy?



Licenciado por la UPM desde el año 2001 y doctor en informática por la UNED desde el año 2009, Javier conjuga sus labores como profesor e investigador con la consultoría y la formación técnica para empresa. Su línea de trabajo actual se centra en la innovación y desarrollo de tecnologías para la Web. Además realiza actividades de evangelización y divulgación en diversas comunidades IT y es coordinador de varios grupos de ámbito local como NodeJS Madrid o Madrid JS. Forma parte del programa Polymer Polytechnic Speaker y es mentor del capítulo de Madrid de Node School.

 javier.velez.reyes@gmail.com

 @javiervelezreye



 [linkedin.com/in/javiervelezreyes](https://www.linkedin.com/in/javiervelezreyes)



 [gplus.to/javiervelezreyes](https://plus.google.com/+javiervelezreyes)

 javiervelezreyes

 youtube.com/user/javiervelezreyes



Javier Vélez Reyes
@javiervelezreye
Javier.velez.reyes@gmail.com

1 *Introducción*

- Arquitectura de Integración
- Programación Asíncrona
- Modelos de Programación Asíncrona

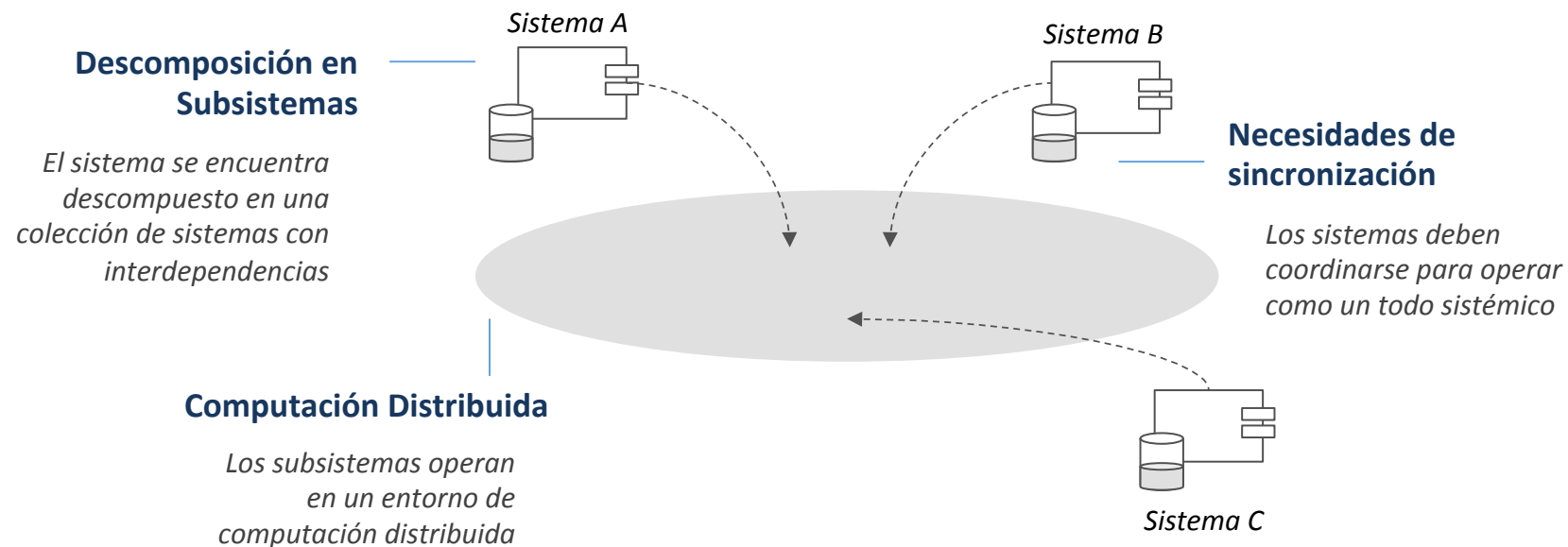
Programación Asíncrona en JavaScript

Introducción

I. Arquitecturas de Integración

¿Qué son las Arquitecturas de Integración?

Las arquitecturas de integración surgen de la necesidad de sincronizar los sistemas de información que operan en un entorno de computación distribuida.



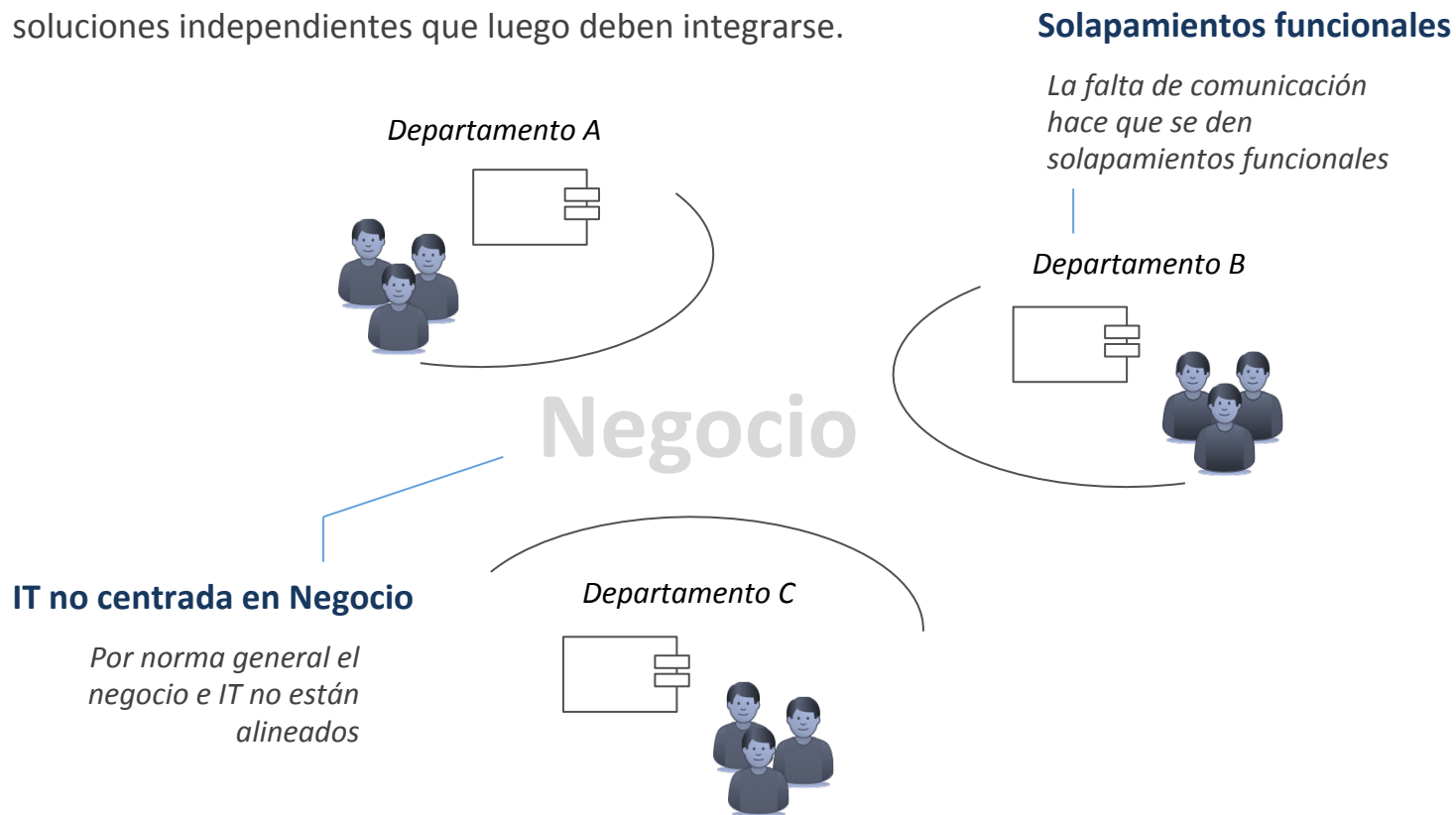
Programación Asíncrona en JavaScript

Introducción

I. Arquitecturas de Integración

¿Por Qué Surgen las Arquitecturas de Integración?

La falta de coordinación entre los equipos de desarrollo hace que el soporte a negocio se construya en base de soluciones independientes que luego deben integrarse.



Programación Asíncrona en JavaScript

Introducción

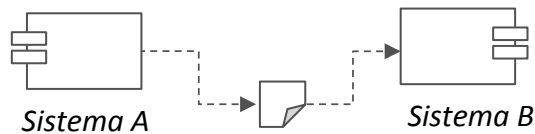
I. Arquitecturas de Integración

Tipos de Arquitecturas de Integración

Se reconocen 4 estilos arquitectónicos que se han venido aplicando recurrentemente en relación a los problemas de integración.

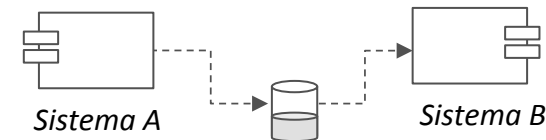
A. Transferencia de Ficheros

1970



1980

B. Bases de Datos Compartidas



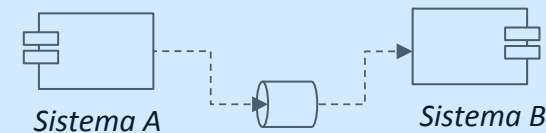
2005

C. Invocación a Procedimientos Remotos



2010

D. Sistemas de Mensajería



Programación Asíncrona en JavaScript

Introducción

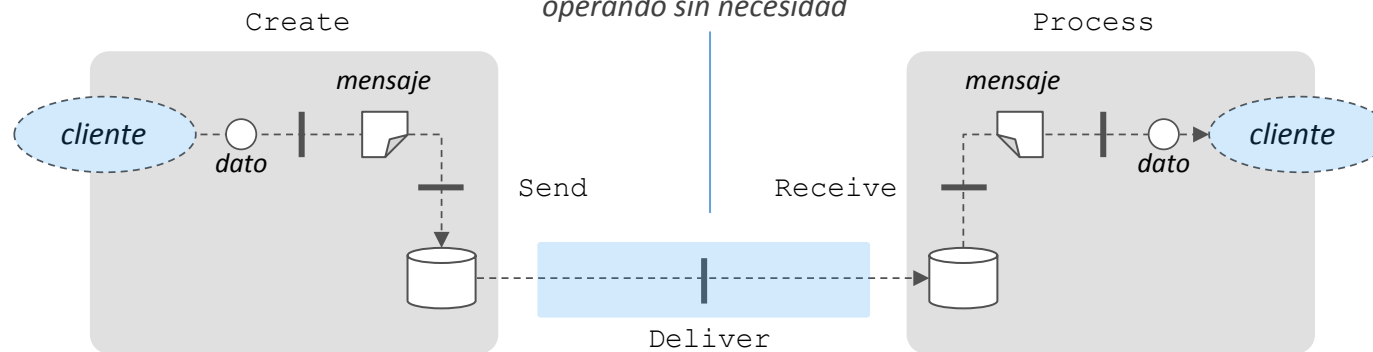
I. Arquitecturas de Integración

Arquitecturas de Integración & Asincronía

Los sistemas de mensajería proporcionan respuestas de alto escalado al hacer las operaciones de acceso al canal no bloqueantes.

Desacoplamiento Temporal

El desacoplamiento temporal permite que los clientes puedan seguir operando sin necesidad



Programación asíncrona

Las operaciones de entrega al canal son no bloqueantes lo que complica la programación de clientes

Programación Asíncrona en JavaScript

Introducción

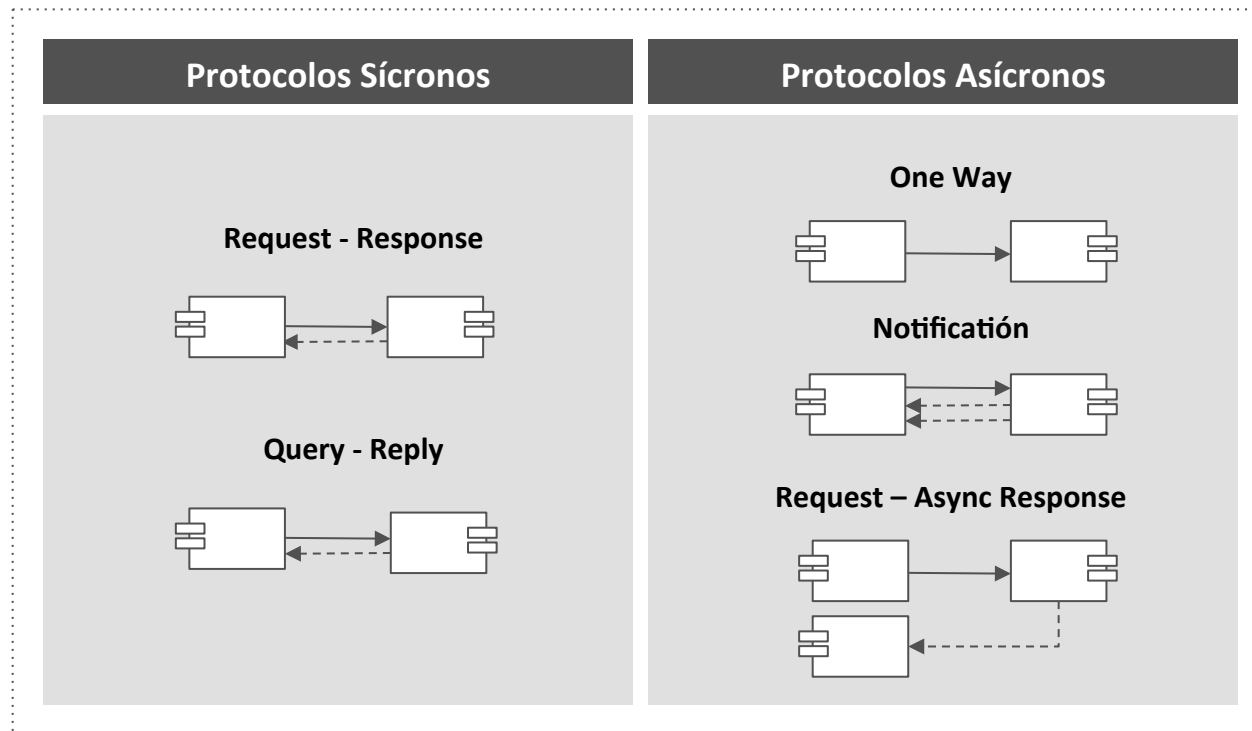
I. Arquitecturas de Integración

Arquitecturas de Integración & Asincronía

Los sistemas de mensajería proporcionan respuestas de alto escalado al hacer las operaciones de acceso al canal no bloqueantes.

Dónde

Medio Asíncrono



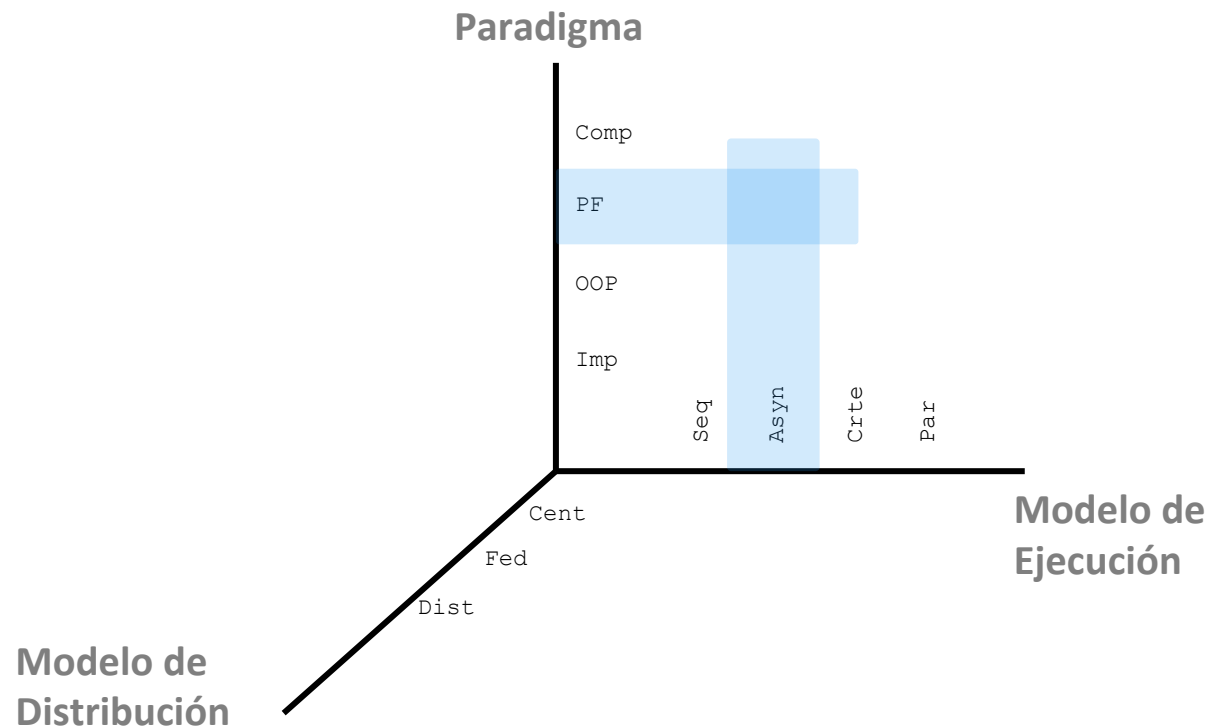
Programación Asíncrona en JavaScript

Introducción

II. Programación Asíncrona

¿Qué es la Programación Asíncrona?

La programación asíncrona es una variante de los modelo de ejecución paralelos que promueve la definición de operaciones no bloqueantes.



Programación Asíncrona en JavaScript

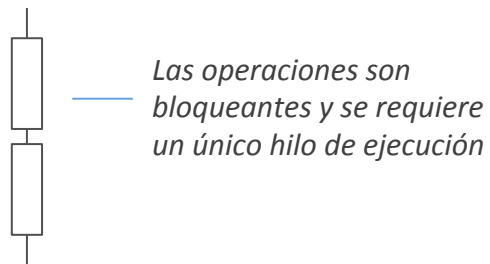
Introducción

II. Programación Asíncrona

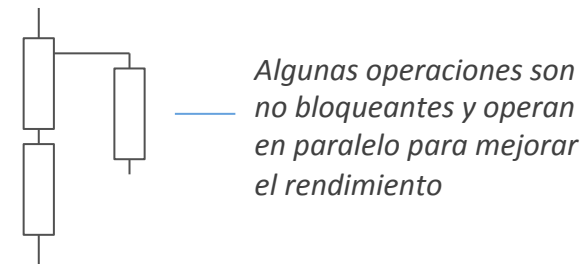
¿Qué No es la Programación Asíncrona?

Conviene diferenciar entre cada uno de los cuatro modelos de ejecución y el tipo de problemas a los que éstos pretenden dar respuesta.

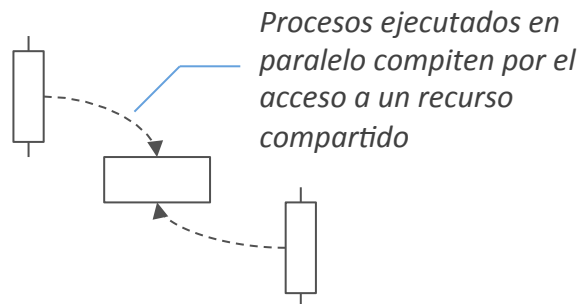
A. Programación Secuencial



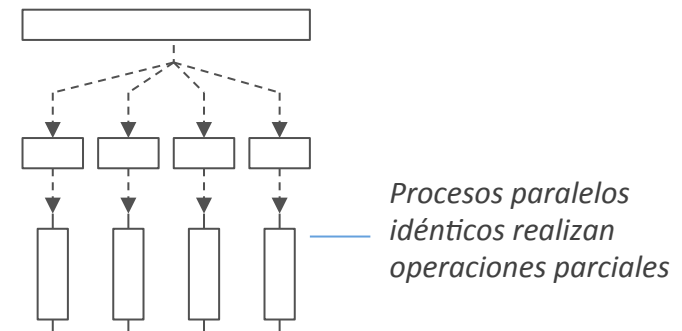
B. Programación Asíncrona



C. Programación Concurrente



D. Programación Paralela



Programación Asíncrona en JavaScript

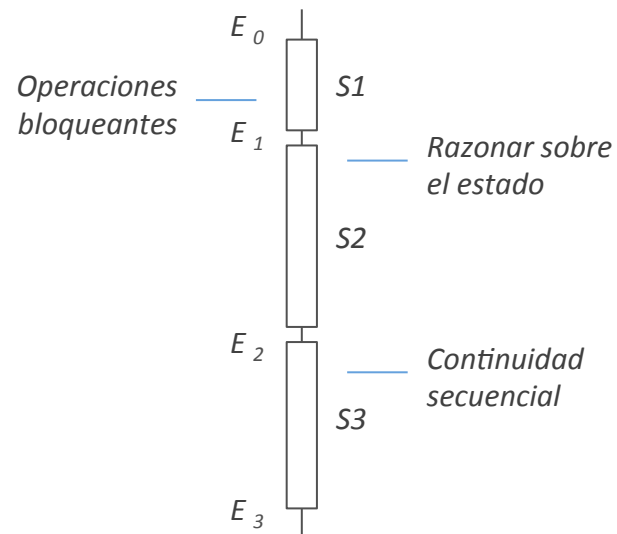
Introducción

II. Programación Asíncrona

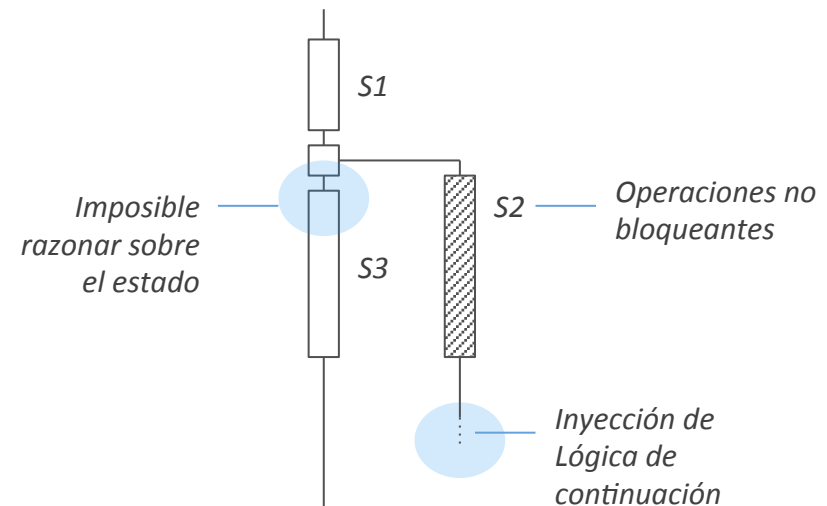
Modelo de Ejecución Síncrona vs Asíncrona

El hecho de introducir la posibilidad de definir funciones no bloqueantes complica considerablemente el razonamiento y la operativa algorítmica.

A. Programación Secuencial



B. Programación Asíncrona



Programación Asíncrona en JavaScript

Introducción

II. Programación Asíncrona

Modelo de Ejecución Asíncrona vs Secuencia

Podemos comprobar que la mayor parte de los constructos sintácticos sobre los que estamos acostumbrados a programar operan sólo en un modelo secuencial.

A. Programación Secuencial

```
✓ return f(...);  
  
✓ try {  
    f(...);  
} catch (e) { ... }  
  
✓ g(f(...));  
  
✓ while (f(...)) {  
    g(...);  
}  
  
✓ if (f(...)) { ... }
```

B. Programación Asíncrona

```
✗ return f(...);  
  
✗ try {  
    f(...);  
} catch (e) { ... }  
  
✗ g(f(...))  
  
✗ while (f(...)) {  
    g(...);  
}  
  
✗ if (f(...)) { ... }
```

Javier Vélez Reyes
@javiervelezreye
Javier.velez.reyes@gmail.com

2 *Modelos de Programación Asíncrona*

- Modelos Basados en Continuaciones
- Modelos Basados en Eventos
- Modelos Basados en Intercesión

Programación Asíncrona en JavaScript
Modelos de Programación Asíncrona

Programación Asíncrona en JavaScript

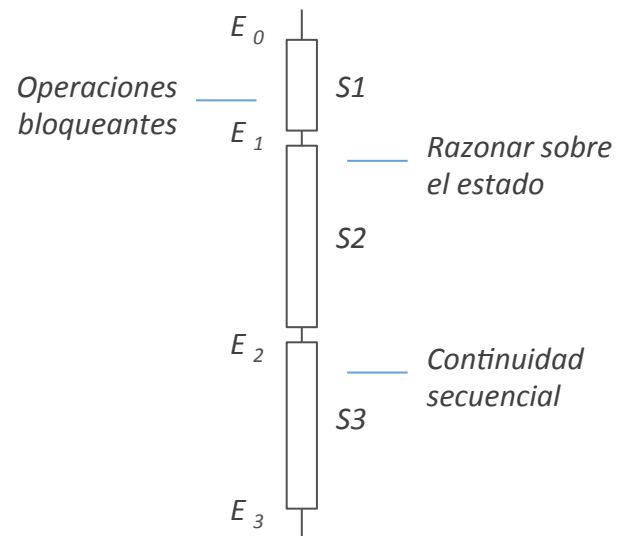
Modelos de Programación Asíncrona

I. Introducción

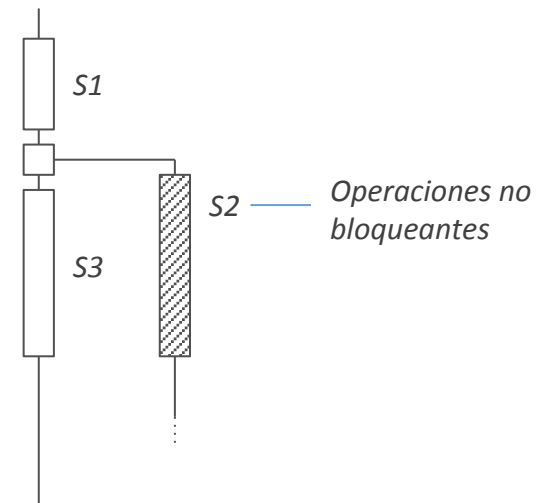
¿Qué son los Modelo de Programación Asíncrona?

La programación asíncrona es un esfuerzo por encontrar mecanismos que simulen la secuencialidad algorítmica a la vez que se mantiene el carácter no bloqueante.

A. Programación Secuencial



B. Programación Asíncrona



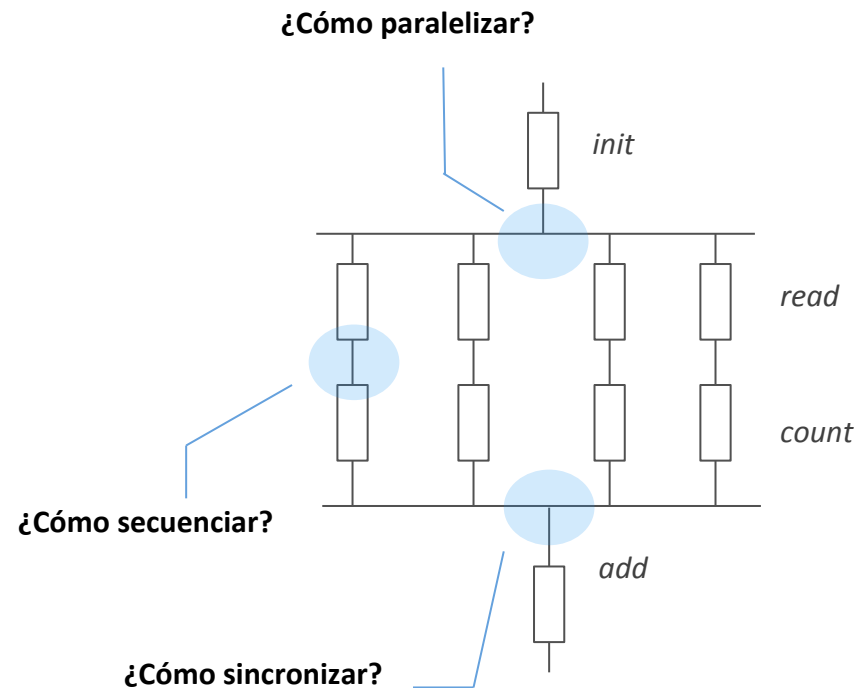
Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

I. Introducción

¿Qué Problemas Esenciales se Trata de Resolver?

Desde una perspectiva reduccionista se puede afirmar que la programación asíncrona discute como modelar la ejecución no bloqueante en secuencia y en paralelo.



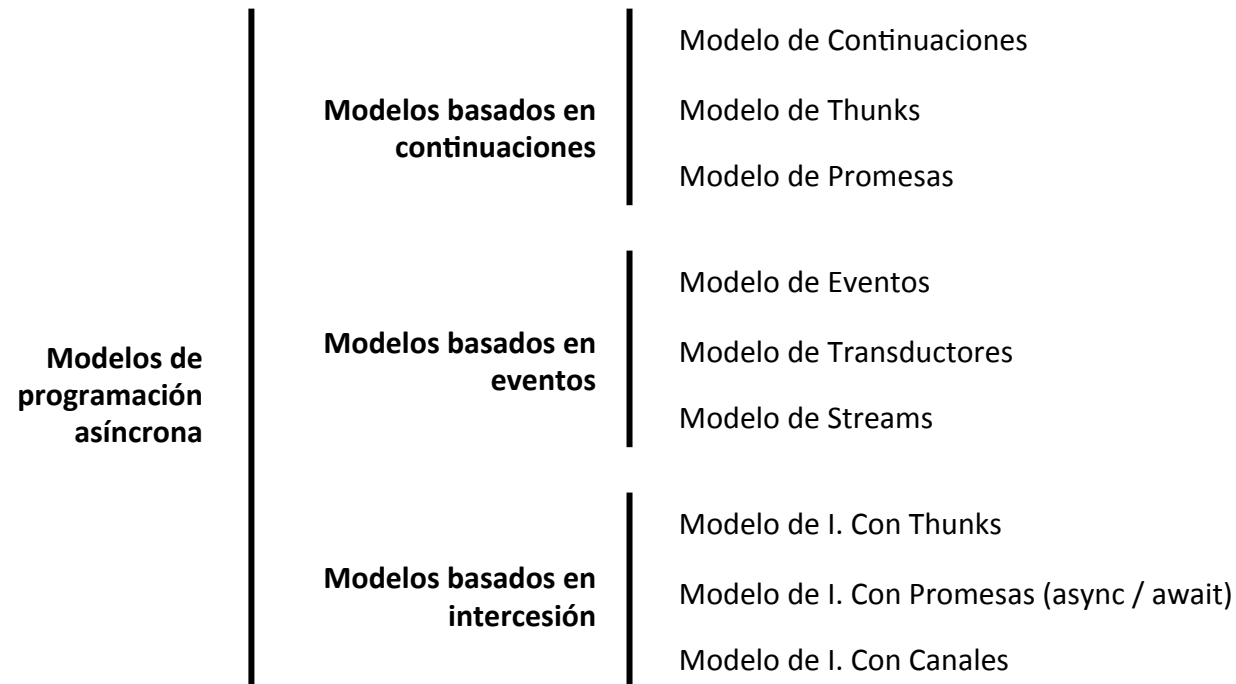
Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

I. Introducción

Tipos de Modelos de Programación Asíncrona

A lo largo de los últimos años se han ideado diferentes modelos de programación asíncrona que pretenden dar respuesta a distintos tipos de escenarios.



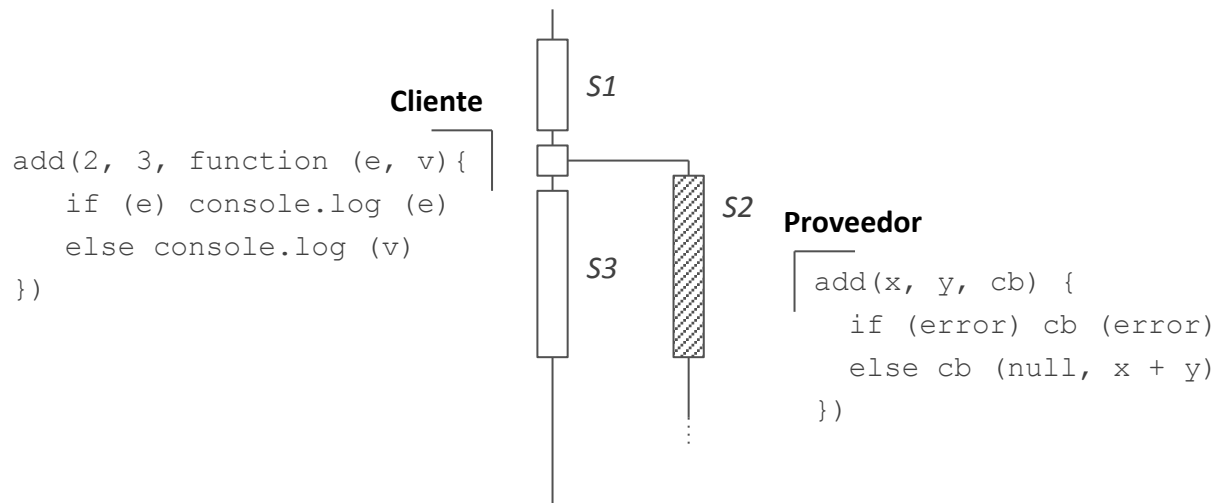
Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

II. Modelos Basados en Continuaciones

Modelo de Paso de Continuaciones

El modelo de paso de continuaciones consiste en aumentar en un parámetro más la aridad de la función para que ésta reciba la lógica de continuación.



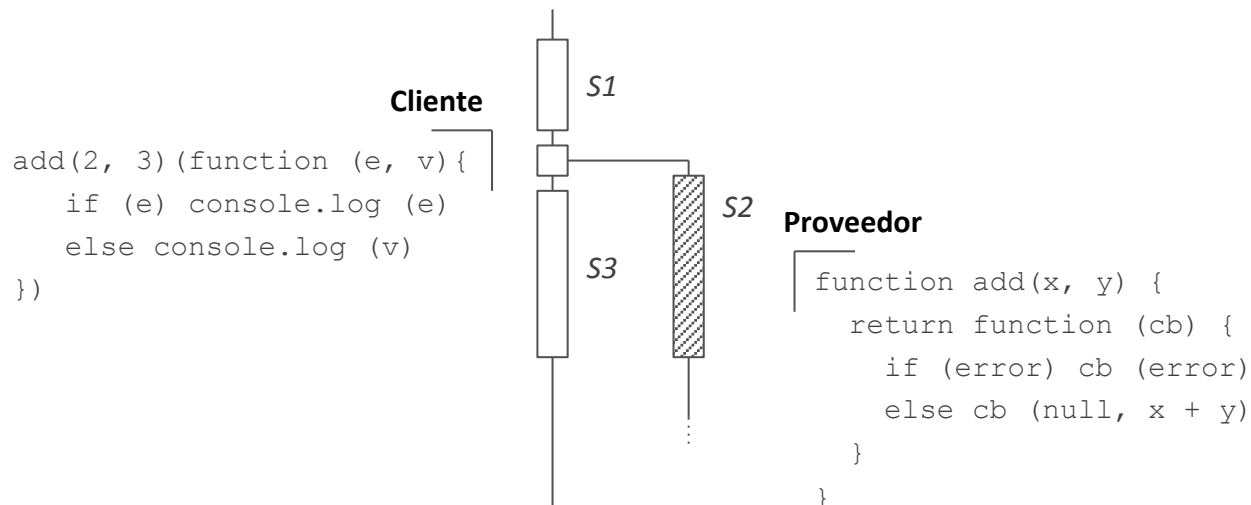
Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

II. Modelos Basados en Continuaciones

Modelo de Thunks

El modelo de paso de continuaciones consiste en aumentar en una fase de evaluación parcial más la aridad de la función para que ésta reciba la lógica de continuación.



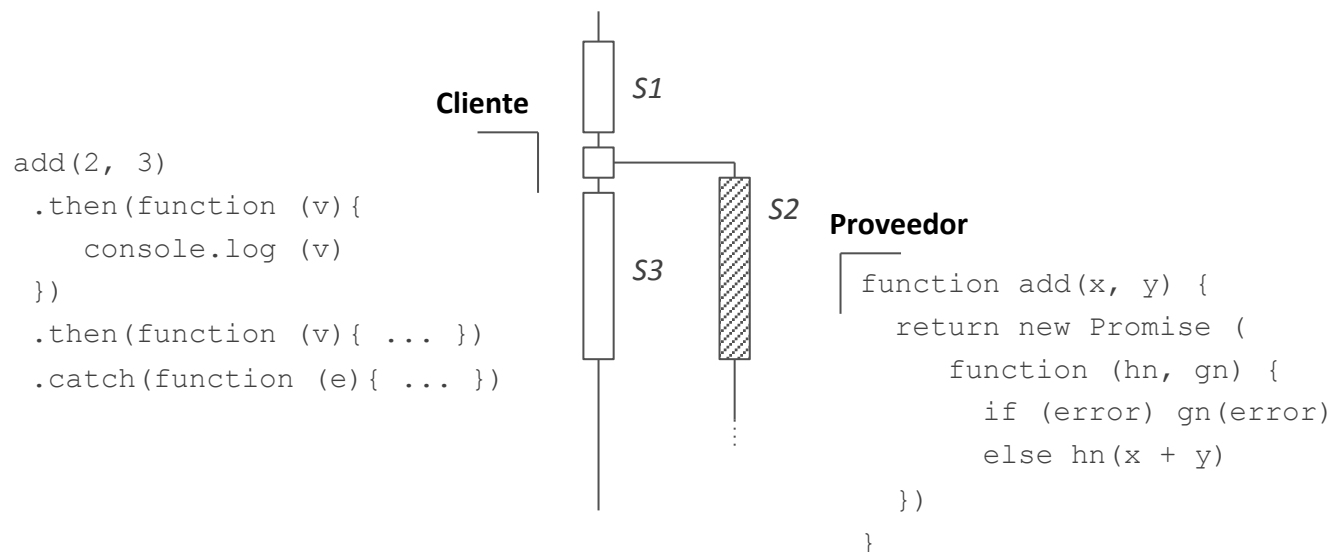
Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

II. Modelos Basados en Continuaciones

Modelo de Promesas

El modelo de promesas genera como respuesta inmediata al cliente una abstracción de datos que representa un compromiso de valor futuro con inyectores para incluir la lógica de continuación.



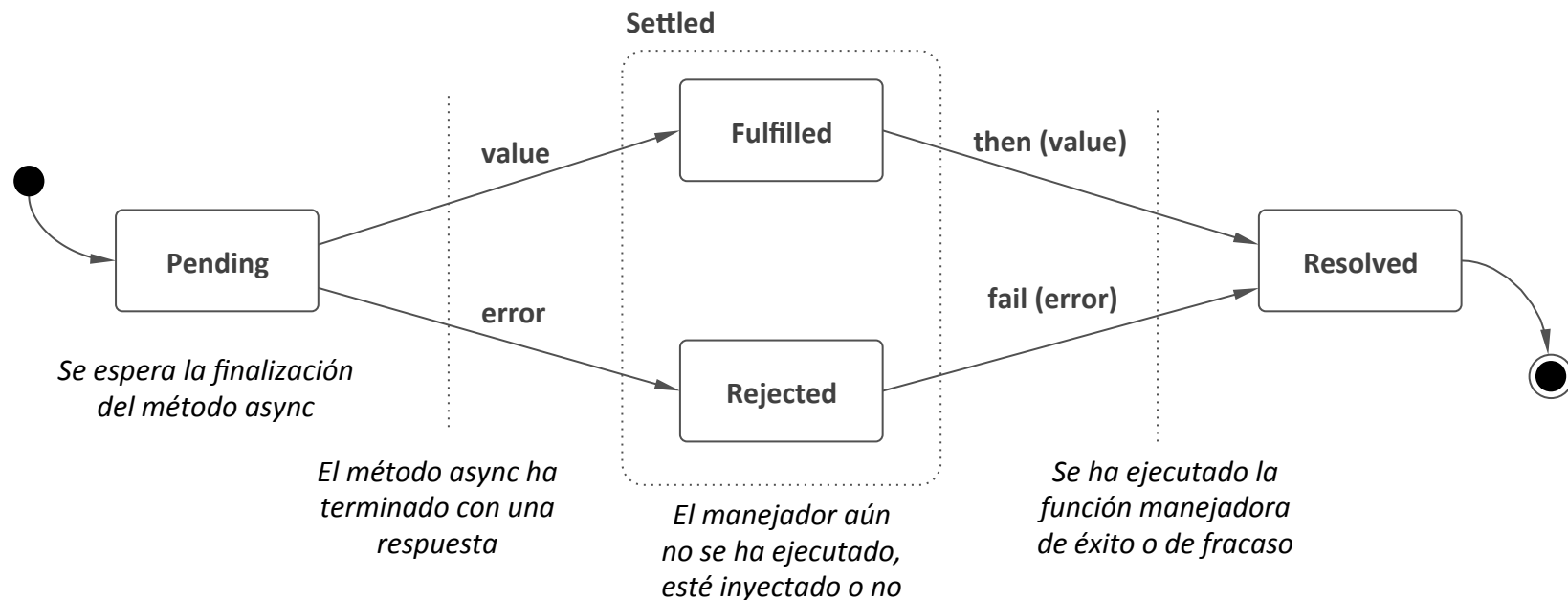
Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

II. Modelos Basados en Continuaciones

Modelo de Promesas

El modelo de promesas genera como respuesta inmediata al cliente una abstracción de datos que representa un compromiso de valor futuro con inyectores para incluir la lógica de continuación.



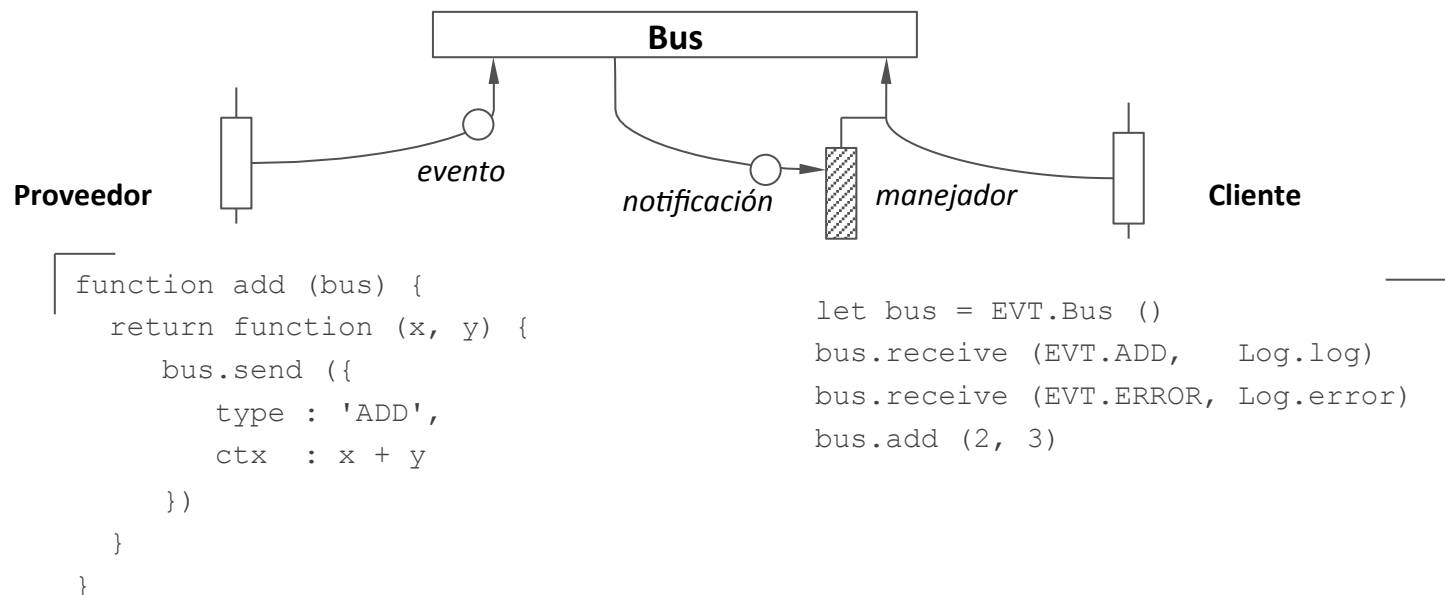
Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

III. Modelos Basados en Eventos

Modelo de Eventos

En el modelo de eventos las operaciones disparan eventos de diferente tipo para que sean escuchados por los manejadores de eventos que los clientes registran en el bus.



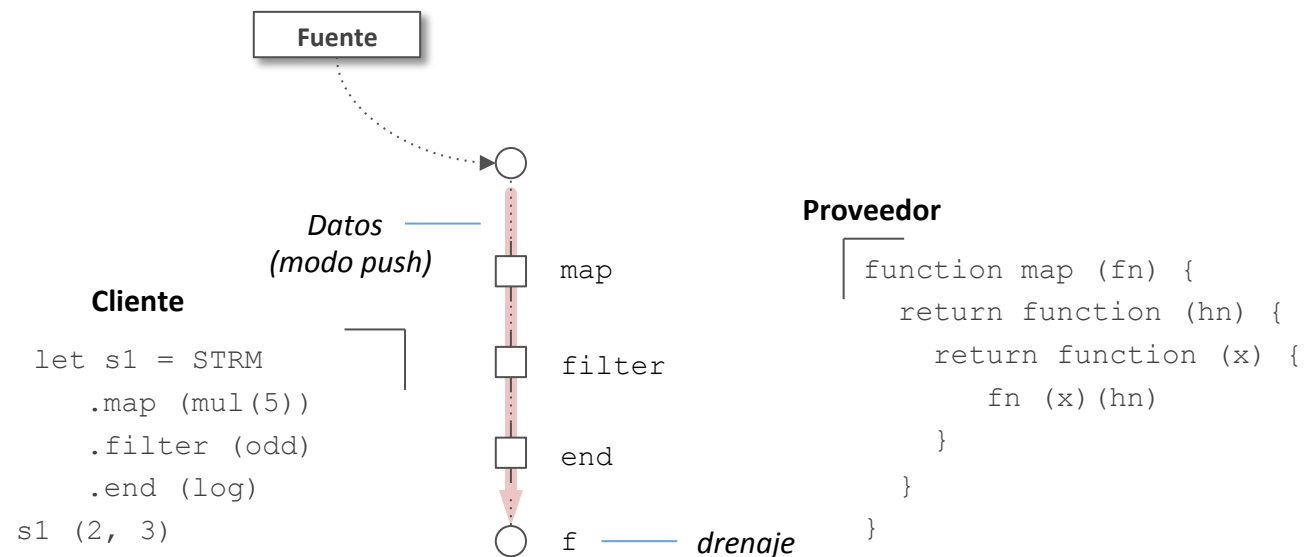
Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

III. Modelos Basados en Eventos

Modelo de Streams

En el modelo de Streams se definen arquitecturas de tuberías por donde fluye la información. Existen modelos de consumición en modo push y modo pull.



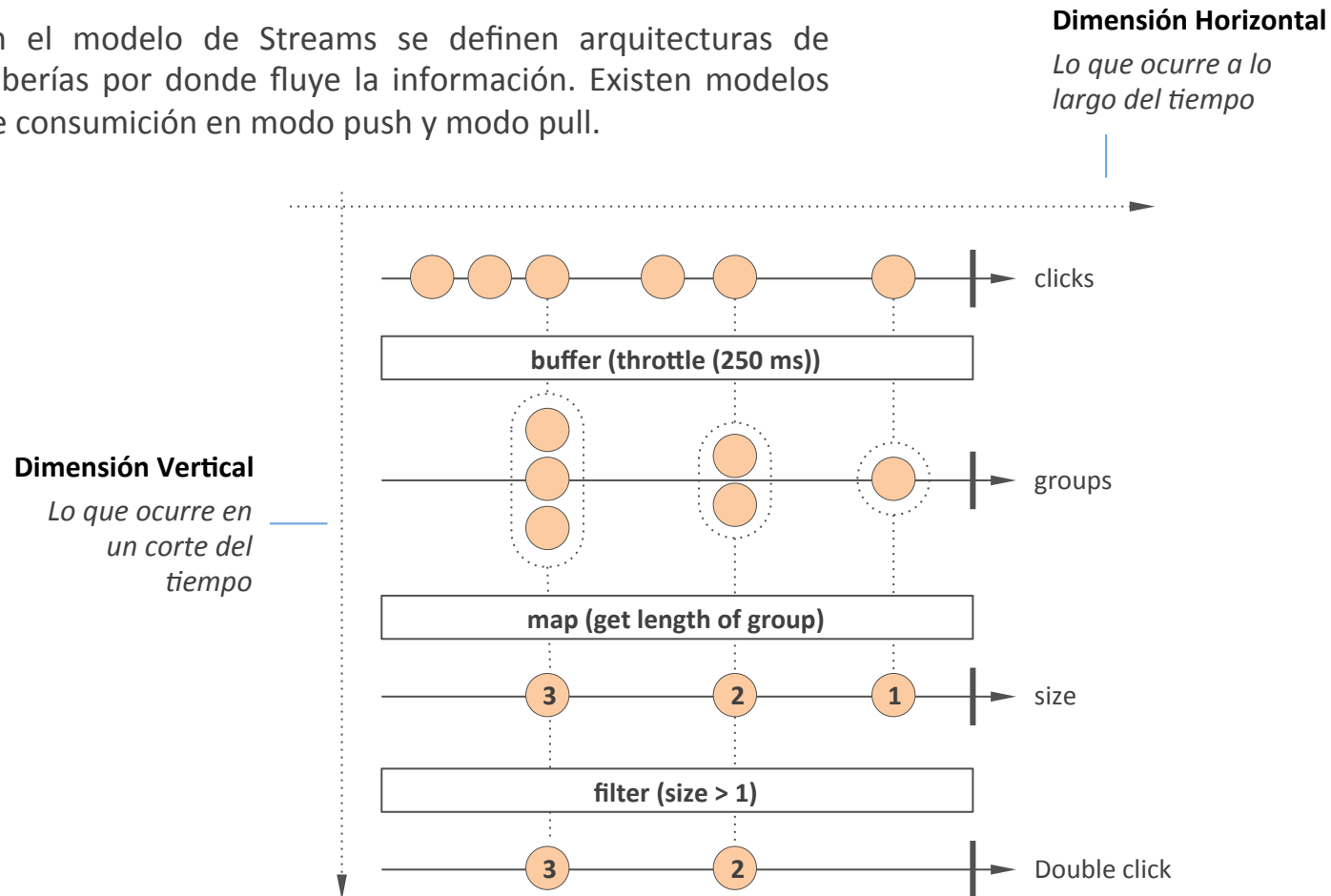
Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

III. Modelos Basados en Eventos

Modelo de Streams

En el modelo de Streams se definen arquitecturas de tuberías por donde fluye la información. Existen modelos de consumición en modo push y modo pull.



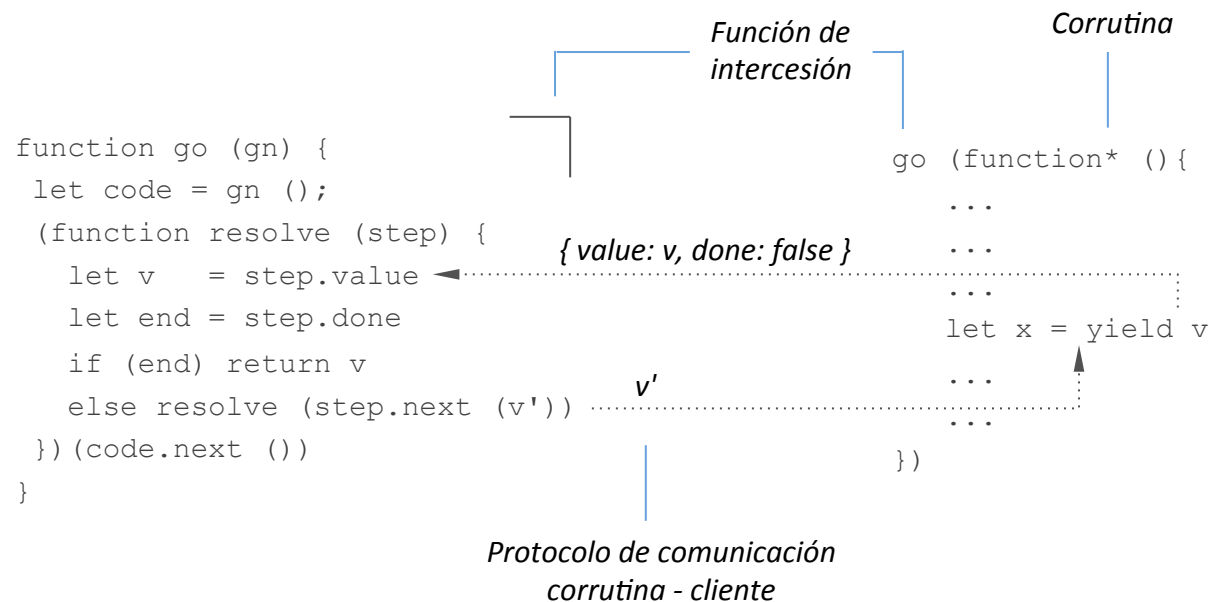
Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

IV. Modelos Basados en Intercesión

Intercesión Basada en Corrutinas

Las corrutinas son una generalización de las rutinas que permiten interrumpir el flujo de ejecución en cualquier momento y devolver el control al cliente para que luego éste vuelva a restaurarlo.



Programación Asíncrona en JavaScript

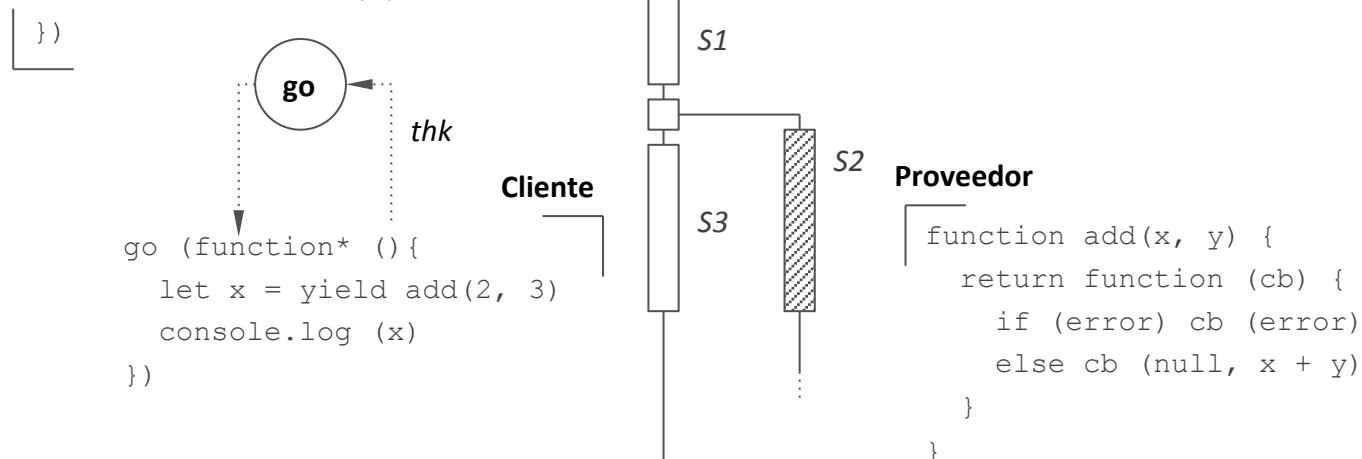
Modelos de Programación Asíncrona

IV. Modelos Basados en Intercesión

Modelo de Co-Thunks

En el modelo de Co-Thunks se pasan thunks a una corrutina que es capaz de enhebrar la lógica de continuación y devolver al contexto de cliente el valor real.

```
thk (function (e, v) {
  if (e) code.throw (e)
  else code.next (v)
})
```



Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

IV. Modelos Basados en Intercesión

Modelo de Co-Promesas

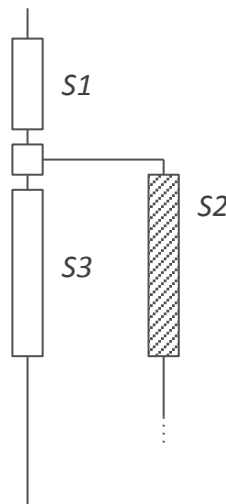
En el modelo de Co-Promesas se pasan promesas a una corrutina que es capaz de enhebrar la lógica de continuación y devolver al contexto de cliente el valor real.

```
p.then (function () {  
  code.next (v)  
}).catch (function (e) {  
  code.throw (e)  
})
```

go

```
go (function* () {  
  let x = yield add(2, 3)  
  console.log (x)  
})
```

Cliente



Proveedor

```
function add(x, y) {  
  return new Promise (  
    function (hn, gn) {  
      if (error) gn(error)  
      else hn(x + y)  
    })  
}
```

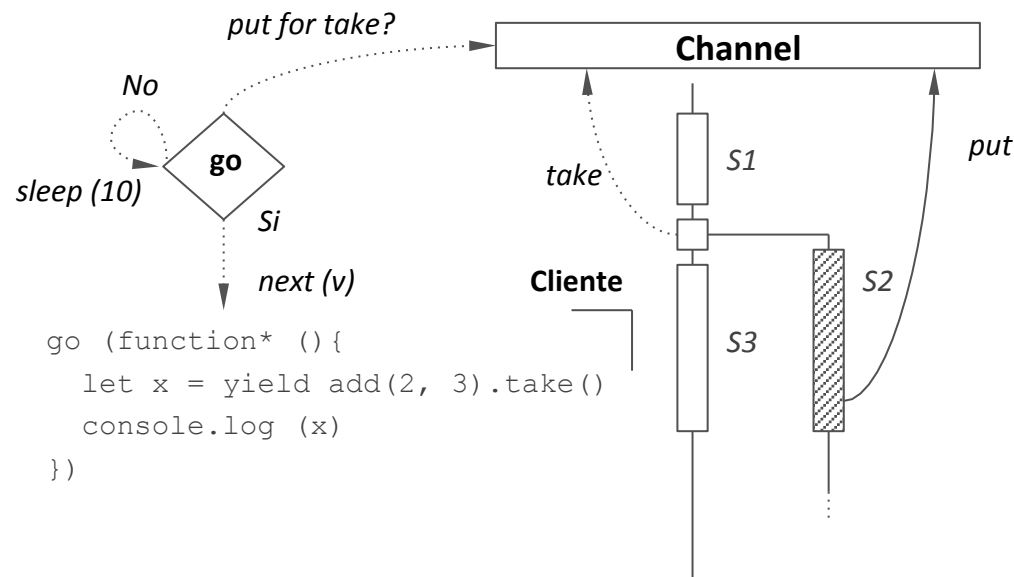
Programación Asíncrona en JavaScript

Modelos de Programación Asíncrona

IV. Modelos Basados en Intercesión

Modelo de Canales

En el modelo de Canales la comunicación entre clientes y proveedores asíncronos se realiza por medio de operaciones de lectura y escritura bloqueante.



Proveedor

```
function add(x, y) {  
  let ch = new Channel ()  
  ch.put (x + y)  
  return ch  
}
```

Programación Asíncrona en JavaScript

Preguntas

Programación Asíncrona en JavaScript

Arquitecturas de Integración



www.javiervelezreyes.com

Programación Asíncrona en JavaScript

Arquitecturas de Integración

Javier Vélez Reyes

@javiervelezreye
Javier.velez.reyes@gmail.com

Diciembre 2016

