Contenido

[1. Autores del trabajo, planificación y entrega 2](#_Toc448254544)

[1.1 Autores 2](#_Toc448254545)

[1.2 Planificación 2](#_Toc448254546)

[1.3 Entrega 2](#_Toc448254547)

[2. Requisitos del prototipo a implementar 3](#_Toc448254548)

[2.1 Requisitos funcionales 3](#_Toc448254549)

[2.2 Otros requisitos 3](#_Toc448254550)

[3. Criterios de comparación en la implementación 4](#_Toc448254551)

[3.1 Criterio 1: Nombre del criterio 4](#_Toc448254552)

[3.2 Criterio 2: Nombre del criterio 4](#_Toc448254553)

[3.N Criterio N: Nombre del criterio 4](#_Toc448254554)

[4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología A 5](#_Toc448254555)

[4.1 Documentación de diseño 5](#_Toc448254556)

[4.2 Documentación de construcción 5](#_Toc448254557)

[4.3 Documentación de pruebas 5](#_Toc448254558)

[4.4 Documentación de instalación 5](#_Toc448254559)

[4.5 Manual de usuario 5](#_Toc448254560)

[5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B 6](#_Toc448254561)

[5.1 Documentación de diseño 6](#_Toc448254562)

[5.2 Documentación de construcción 6](#_Toc448254563)

[5.3 Documentación de pruebas 6](#_Toc448254564)

[5.4 Documentación de instalación 6](#_Toc448254565)

[5.5 Manual de usuario 6](#_Toc448254566)

[6. Comparación de las dos implementaciones 7](#_Toc448254567)

[6.1 Evaluación de los criterios en la implementación usando la tecnología A 7](#_Toc448254568)

[6.2 Evaluación de los criterios en la implementación usando la tecnología B 7](#_Toc448254569)

[7. Comparación de la implementación de las tecnologías 8](#_Toc448254570)

[8. Conclusiones 10](#_Toc448254571)

# 5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología B, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

## 5.1 Documentación de diseño

Hay que incluir la descripción del diseño del prototipo, incluyendo diagramas, y el diseño de la interfaz de usuario.

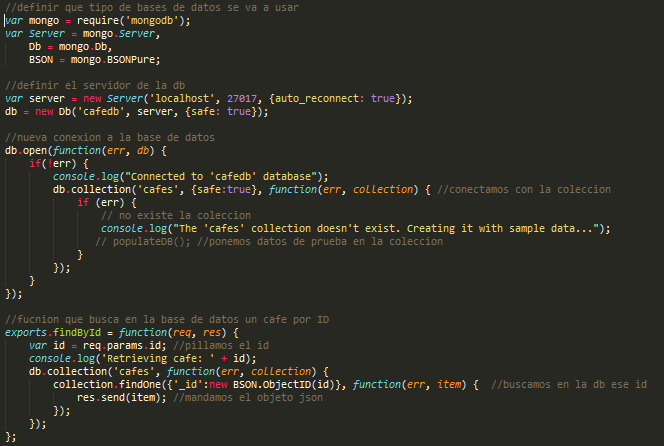
## 5.2 Documentación de construcción

Se ha implementado Node.js utilizando Express, Bootstrap y MongoDB.

Es por esto por lo que el proyecto cuenta con distintos archivos:

* **Routes/cafes.js:**

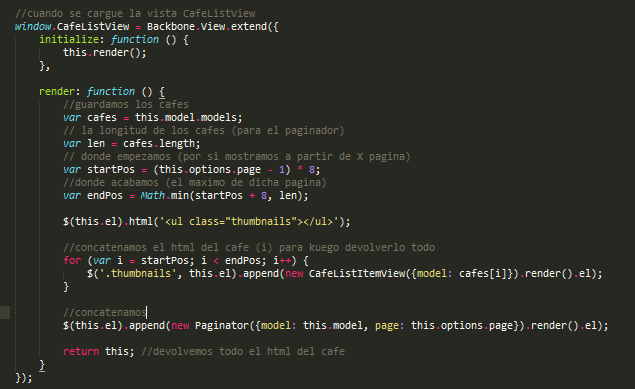
Módulo js encargado de lo relacionado con la base de datos Mongo, actualizar datos, borrarlos, enviarlos, etc.



* **Public/js/views:**

En js/views se encuentran los archivos js dedicados al manejo de las vistas. Aquí podemos ver el controlador que se ocupa de mostrar la lista de cafés al inicio.

* + **cafelist.js:**



* **Public/js/main.js**

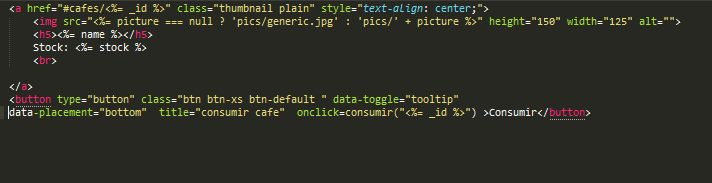
Este código es el encargado de saber que vista cargar según se indique en el navegador.



* **Public/tpl**

Aquí se pueden encontrar el código html de las vistas (About, lista de cafés, vista individual de un café y navbar).

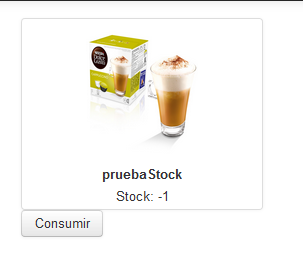
* + **CafeListItemView.html:**



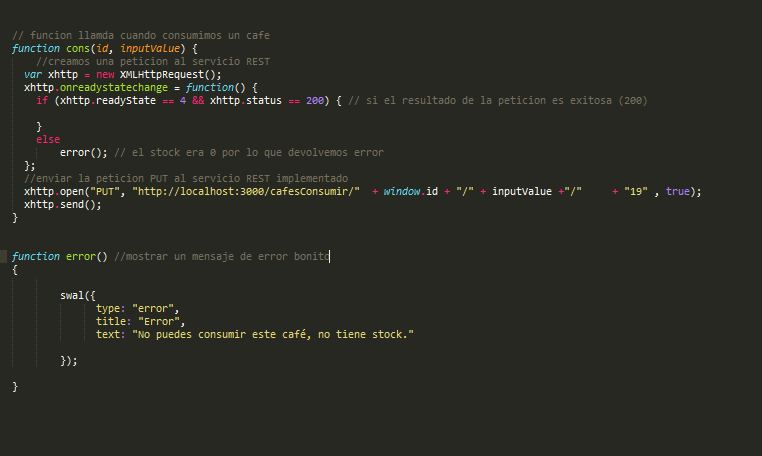
## 5.3 Documentación de pruebas

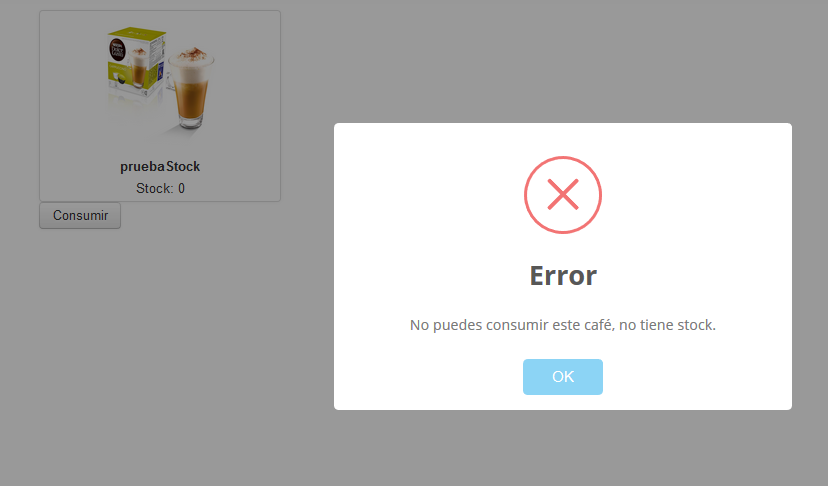
* Corrección del stock:

Cuando el stock llegaba a 0, la aplicación seguía restando:



Para corregirlo se implementó un mensaje de error cuando el stock es 0:





* El stock contiene letras:

Este error se da cuando al crear un café añades letras, por ejemplo: “cinco”.

Nuestro prototipo actualmente no tiene implementado ninguna solución a este problema.

## 5.4 Documentación de instalación

Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.

## 5.5 Manual de usuario

Descripción suficiente para que una persona que no ha participado en el proyecto pueda utilizar toda la funcionalidad que ofrece el prototipo. Que debe coincidir con los requisitos funcionales incluidos en el apartado 2.

# 8. Conclusiones

Tras la realización de ambos prototipos del inventario de cafés, se llega a la conclusión de que ambos sistemas son ciertamente potentes y útiles, pero se han encontrado ciertas desventajas y ventajas que serán comentadas a continuación.

**Nodejs** tiene una gran facilidad de comunicación con los navegadores web y con las bases de datos como Mongo, pero el hecho de tratarse de un lenguaje nuevo en el lado del servidor hace que falten librerías generales.

El código generado no se encuentra correctamente organizado, como se puede ver en las tablas generadas de líneas de código, el proyecto de nodejs cuenta con multitud de archivos, subcarpetas y sitios donde es necesario tocar aun siendo esta una aplicación muy simple. Haciendo que la mantenibilidad del código sea una ardua tarea.

**Django** por otro lado no cuenta con prácticamente ninguna de las desventajas mencionadas anteriormente, es elegante y flexible, ordenado y sobre todo simplificado.

Las desventajas de django son que al tratarse de un framework viejo, evoluciona lento en comparación con nodejs. Al basarse en Python, los servidores con dicho hosting son más escasos.

En general, django es más práctico y fácil de usar, por lo que ante un proyecto general, nos decantamos antes por dicho framework.