

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Análisis y Diseño 1

Sección N

Ing. Jose Manuel Ruiz Juarez

Aux. César Dionicio Sazo Mayen

Proyecto 2
Planificación de pruebas

All for one - Grupo No. 7	
Nombre	Carnet
Erick Fernando Sánchez Mejía	201503878
Melyza Alejandra Rodriguez Contreras	201314821
Helmut Efraín Najarro Álvarez	201712350
Christopher Alexander Acajabon Gudiel	201404278

Planificación de pruebas

Equipo de trabajo

Miembro		Rol que desempeña
	Erick Fernando Sánchez Mejía 201503878	Service Request Manager Desarrollador Frontend 1
	Helmut Efraín Najarro Álvarez 201712350	Service Delivery Manager Desarrollador Frontend 2
	Christopher Alexander Acajabon Gudiel 201404278	Desarrollador Backend 1
	Melyza Alejandra Rodriguez Contreras 201314821	Desarrollador Backend 2

Documentación de pruebas

Pruebas unitarias

Módulos a ser probados	Los módulos sometidos a pruebas unitarias son los siguientes: <ul style="list-style-type: none">• Usuarios• Productos• Carrito de compras• Compra• Facturación
Objetivos de las pruebas	En estos módulos se realizarán pruebas para validar los siguientes puntos:

	<ul style="list-style-type: none"> • Aplicación de reglas de registro de usuarios. • Validación de datos ingresados por el usuario para registrarse. • Persistencia del registro de usuario. • Validación de credenciales de usuario. • Obtención de datos de productos. • Manejo de stock de productos. • Manejo de productos en carrito de compras. • Flujo de una compra. • Lógica de generación de facturas. • Flujo de sesión de usuario.
Orden de las pruebas	<p>Los módulos se probarán independientemente, siguiendo el orden detallado a continuación para garantizar su continuidad.</p> <ul style="list-style-type: none"> • Usuarios • Productos • Carrito de compras • Compra • Facturación
Responsabilidad de las pruebas	<p>Las pruebas son responsabilidad de los Desarrolladores Backend del proyecto, quienes tendrán a su cargo diseñar el código que permita ejecutar las pruebas que aseguren la efectividad del sistema.</p>

• Seguridad

Vulnerabilidades						
Categoría de prueba	ALTA		MEDIA		BAJA	
C: Corregido P: Pendiente	C	P	C	P	C	P
Web Site Pilfering						
Modifying inputs and Parameter Tampering				1		
Bypassing client side validation		1				
Cookie Abuse						
URL Jumping						

Web Site Pilfering

- Comentarios innecesarios y reveladores (**no encontrados**)
- Direcciones IP (**no encontrados**)
- Direcciones de correo electrónico (**no encontrados**)
- Strings de conexión de base de datos (**no encontrados**)
- Campos escondidos (**no encontrados**)

Modifying inputs and Parameter Tampering

Vulnerabilidad en las facturas	
Riesgo	MEDIO
Complejidad	BAJO
<p>Descripción:</p> <p>La aplicación provee la función de enviar correos a los usuario luego de realizar una compra, en dicho correo se envía un enlace en donde se puede consultar la factura y descargarla en formato PDF. Esto posee una vulnerabilidad ya que para que el usuario pueda ver su factura, necesita ingresar a la siguiente dirección: http:HOST/ticket/numeroTicket siendo el numeroTicket el número de la factura. Si se reemplaza ese número con cualquier otro, mostrará la factura sin ningún problema.</p>	

Bypassing client side validation

Vulnerabilidad en las compras	
Riesgo	ALTO
Complejidad	BAJO
<p>Descripción:</p> <p>La aplicación registra compras a través de peticiones POST, en donde el cuerpo indica los datos del usuario y los de los productos a comprar. Por medio de aplicaciones que simulen las peticiones, como lo es Postman, se pueden registrar compras sin necesidad de autenticación de métodos de pago.</p>	

Cookie Abuse

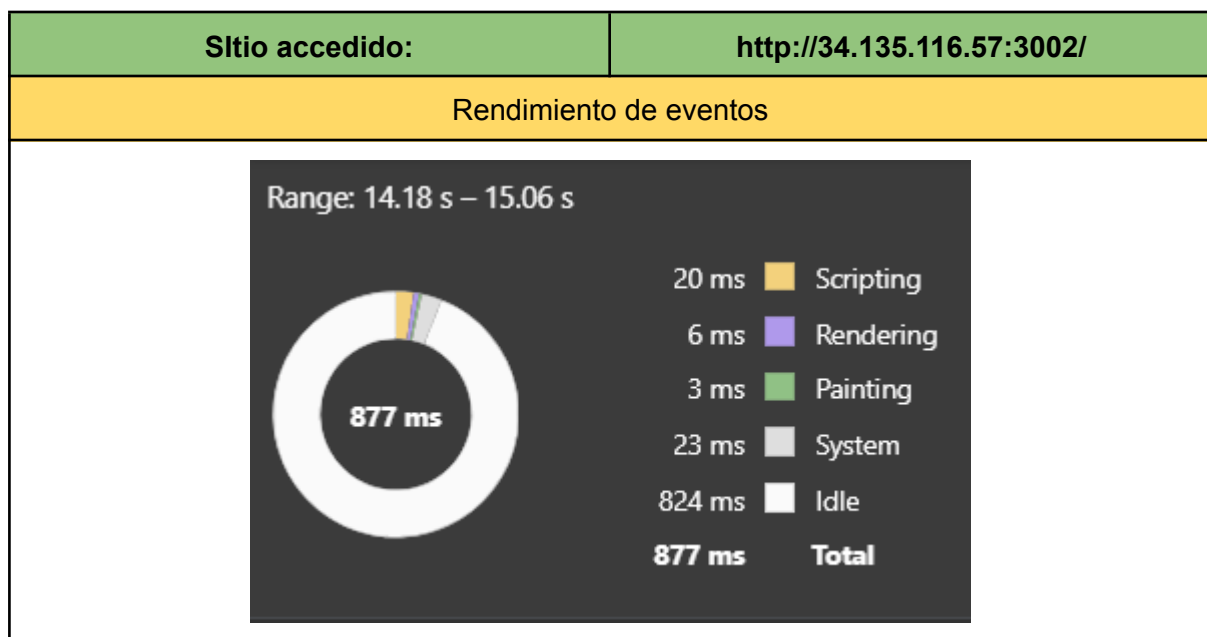
No se utiliza almacenamiento por cookies

URL Jumping

Página	Dirección	Éxito	Fracaso
Inicio	/		1
Productos	/products/categoria		1
Carrito	/cart		1

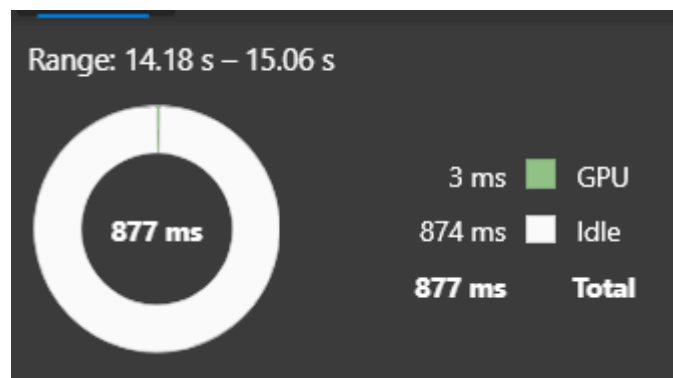
Ninguna de las páginas puede ser accedida sin autenticación.

- Rendimiento



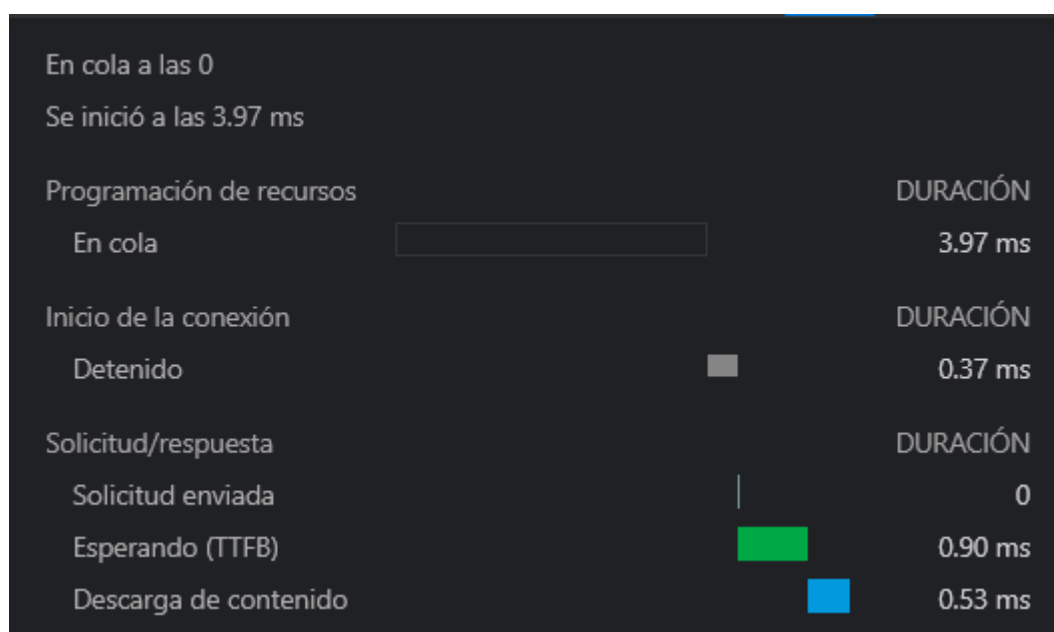
Self Time	Total Time	Activity
7.0 ms 23.5 %	20.3 ms 67.8 %	Event: scroll
3.1 ms 10.3 %	3.1 ms 10.3 %	Update Layer Tree
2.4 ms 8.0 %	2.4 ms 8.0 %	Hit Test
2.2 ms 7.5 %	2.2 ms 7.5 %	Composite Layers
1.1 ms 3.5 %	1.1 ms 3.5 %	Paint
0.5 ms 1.5 %	0.5 ms 1.5 %	Recalculate Style
0.4 ms 1.2 %	0.4 ms 1.2 %	Scroll

Uso de GPU



Rendimiento de Red

Vista general de la página



■ Network request

URL 34.135.116.57/

Duration 177.49 ms (84.84 ms network transfer + 92.65 ms resource loading)

Request Method GET

Priority Máximo

Mime Type text/html

Encoded Data 181 B

Decoded Body 3.1 kB

Uso de API /obtenerProductos

En cola a las 189.30 ms

Se inició a las 190.31 ms

Programación de recursos

DURACIÓN

En cola

1.01 ms

Inicio de la conexión

DURACIÓN

Detenido

0.85 ms

Solicitud/respuesta

DURACIÓN

Solicitud enviada

0.20 ms

Esperando (TTFB)

128.31 ms

Descarga de contenido

184.68 ms

■ Network request

URL :3200/obtenerProductos

Duration 369.63 ms (353.17 ms network transfer + 16.46 ms resource loading)

Request Method GET

Priority Alto

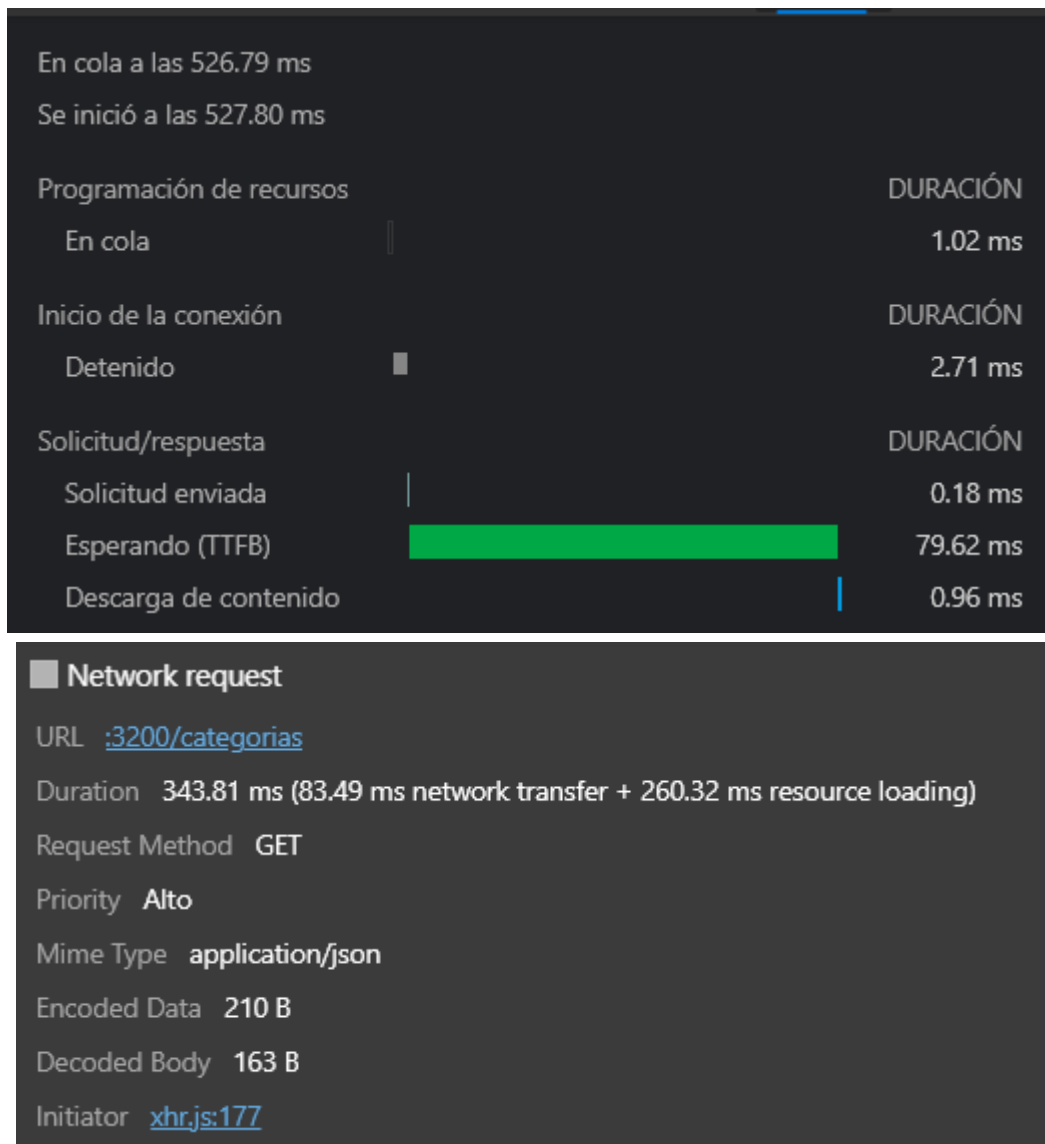
Mime Type application/json

Encoded Data 61.0 kB

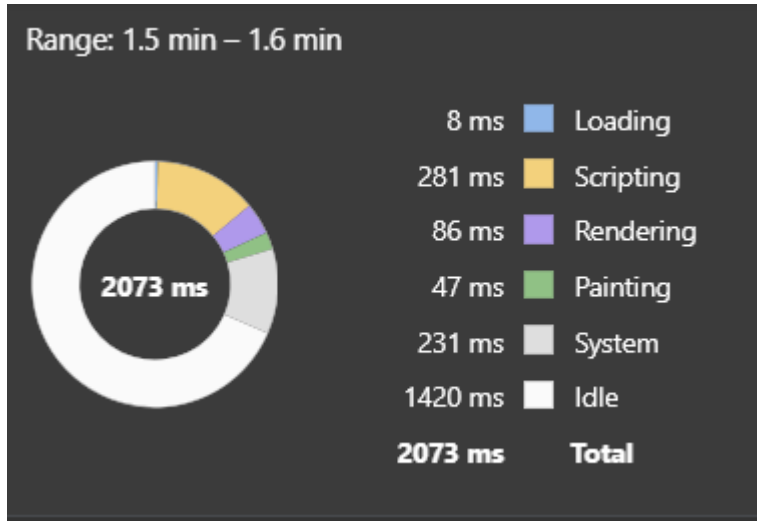
Decoded Body 60.7 kB

Initiator [xhr.js:177](#)

Uso de API /categorias

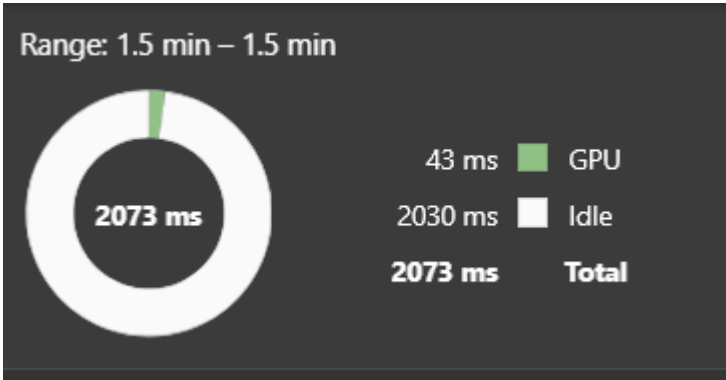


Sitio accedido	http://34.135.116.57:3002/login
Rendimiento de eventos	



Self Time		Total Time		Activity
7.8 ms	1.8 %	148.7 ms	35.2 %	▶ Parse HTML
22.0 ms	5.2 %	147.2 ms	34.8 %	▶ Event: keydown
35.8 ms	8.5 %	38.3 ms	9.1 %	▶ Paint
27.5 ms	6.5 %	27.5 ms	6.5 %	▶ Update Layer Tree
0.9 ms	0.2 %	20.1 ms	4.8 %	▶ XHR Ready State Change
19.8 ms	4.7 %	19.8 ms	4.7 %	▶ Recalculate Style
11.1 ms	2.6 %	11.3 ms	2.7 %	▶ Composite Layers
3.4 ms	0.8 %	3.4 ms	0.8 %	▶ Layout
2.9 ms	0.7 %	2.9 ms	0.7 %	▶ Hit Test
1.0 ms	0.2 %	1.9 ms	0.4 %	▶ Function Call
0.0 ms	0.0 %	0.6 ms	0.1 %	▶ (anonymous)
0.5 ms	0.1 %	0.5 ms	0.1 %	▶ Fire Idle Callback
0.2 ms	0.1 %	0.2 ms	0.1 %	▶ Run Microtasks
0.1 ms	0.0 %	0.1 ms	0.0 %	▶ XHR Load

Uso de GPU



Rendimiento de Red

Vista general de la página

■ Network request

URL [:3002/login](#)

Duration 126.52 ms (89.55 ms network transfer + 36.97 ms resource loading)

Request Method GET

Priority Máximo

Mime Type text/html

Encoded Data 3.3 kB

Decoded Body 3.1 kB

usando la API /login

■ Network request

URL [:3200/login](#)

Duration 266.03 ms (259.46 ms network transfer + 6.57 ms resource loading)

Request Method POST

Priority Alto

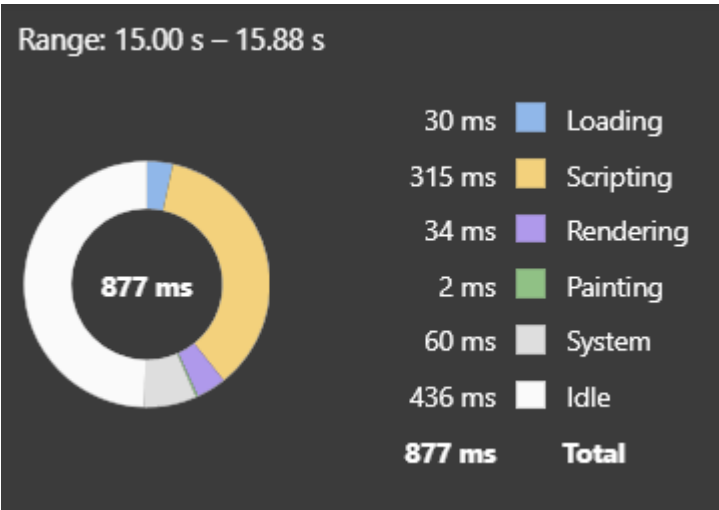
Mime Type application/json

Encoded Data 418 B

Decoded Body 150 B

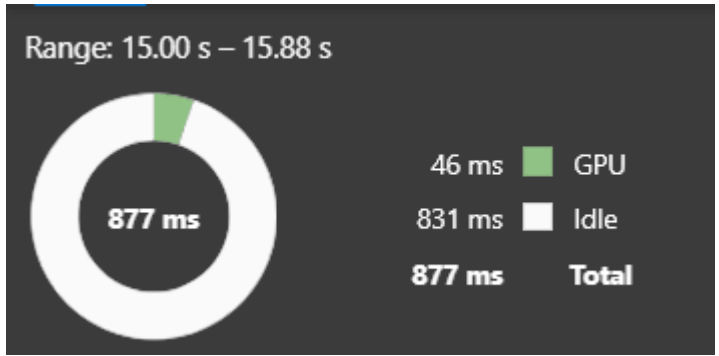
Initiator [xhr.js:177](#)

Sitio accedido	http://34.135.116.57:3002/products/idCategoria
Rendimiento de eventos	



Self Time		Total Time		Activity
8.8 ms	2.3 %	335.4 ms	88.0 %	▶ Parse HTML
24.1 ms	6.3 %	24.2 ms	6.3 %	▶ Layout
5.6 ms	1.5 %	14.5 ms	3.8 %	▶ Event: pointerdown
1.5 ms	0.4 %	1.5 ms	0.4 %	▶ Paint
1.3 ms	0.4 %	1.3 ms	0.4 %	▶ Update Layer Tree
1.1 ms	0.3 %	1.3 ms	0.3 %	▶ Function Call
0.9 ms	0.2 %	1.1 ms	0.3 %	▶ Composite Layers
0.0 ms	0.0 %	1.0 ms	0.3 %	▶ (anonymous)
0.5 ms	0.1 %	0.5 ms	0.1 %	▶ Recalculate Style
0.4 ms	0.1 %	0.4 ms	0.1 %	▶ Hit Test

Uso de GPU



Rendimiento de Red

Vista general de la página

■ Network request

URL [:3002/products/2](#)

Duration 162.70 ms (128.99 ms network transfer + 33.71 ms resource loading)

Request Method GET

Priority **Máximo**

Mime Type text/html

Encoded Data 181 B

Decoded Body 3.1 kB

haciendo uso de la API `/filtroCategoria/idCategoria`

■ Network request

URL [:3200/filtroCategoria/2](#)

Duration 369.19 ms (301.40 ms network transfer + 67.79 ms resource loading)

Request Method GET

Priority **Alto**

Mime Type application/json

Encoded Data 212 B

Decoded Body 13.7 kB

Initiator [xhr.js:177](#)

haciendo uso de la API `/categorias`

■ Network request

URL [:3200/categorias](#)

Duration 183.55 ms (164.70 ms network transfer + 18.85 ms resource loading)

Request Method GET

Priority **Alto**

Mime Type application/json

Encoded Data 210 B

Decoded Body 163 B

Initiator [xhr.js:177](#)

haciendo uso de la API /agregarProductoCarrito

■ Network request

URL :3200/agregarProductoCarrito/

Duration 1.56 s (183.69 ms network transfer + 1.38 s resource loading)

Request Method POST

Priority Alto

Mime Type text/html

Encoded Data 266 B

Decoded Body 8 B

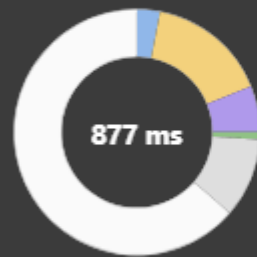
Initiator xhr.js:177

Sitio accedido

http://34.135.116.57:3002/cart

Rendimiento de eventos

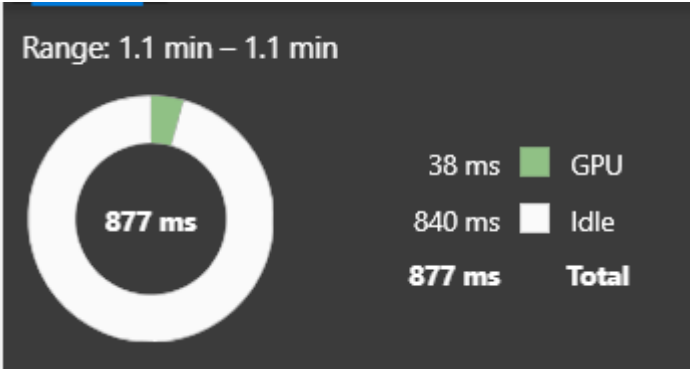
Range: 1.1 min – 1.1 min



27 ms	■	Loading
139 ms	■	Scripting
53 ms	■	Rendering
9 ms	■	Painting
91 ms	■	System
558 ms	■	Idle
877 ms		Total

Self Time	Total Time	Activity
6.2 ms 2.7 %	109.8 ms 48.3 %	▶ Parse HTML
0.3 ms 0.1 %	39.1 ms 17.2 %	▶ XHR Ready State Change
28.6 ms 12.6 %	28.6 ms 12.6 %	Layout
7.2 ms 3.2 %	26.3 ms 11.6 %	▶ Event: pointermove
7.5 ms 3.3 %	7.5 ms 3.3 %	Recalculate Style
7.3 ms 3.2 %	7.3 ms 3.2 %	Paint
4.6 ms 2.0 %	4.6 ms 2.0 %	Update Layer Tree
2.1 ms 0.9 %	2.1 ms 0.9 %	Hit Test
1.4 ms 0.6 %	1.5 ms 0.7 %	▶ Composite Layers
0.0 ms 0.0 %	0.4 ms 0.2 %	▶ (anonymous)
0.2 ms 0.1 %	0.2 ms 0.1 %	Function Call
0.0 ms 0.0 %	0.0 ms 0.0 %	XHR Load

Uso de GPU



Rendimiento de Red

Vista general de la página

■ Network request

URL [:3002/cart/21](#)

Duration 127.36 ms (93.52 ms network transfer + 33.84 ms resource loading)

Request Method GET

Priority Máximo

Mime Type text/html

Encoded Data 181 B

Decoded Body 3.1 kB

Usando API /leerCarrrito/idUsuario

■ Network request

URL [:3200/leerCarrrito/21](#)

Duration 130.76 ms (90.14 ms network transfer + 40.63 ms resource loading)

Request Method GET

Priority Alto

Mime Type application/json

Encoded Data 1.3 kB

Decoded Body 1.0 kB

Initiator [xhr.js:177](#)

Usando API /actualizarCarrito

■ Network request

URL [:3200/actualizarCarrito](#)

Duration 182.44 ms (177.57 ms network transfer + 4.88 ms resource loading)

Request Method POST

Priority Alto

Mime Type text/html

Encoded Data 262 B

Decoded Body 4 B

Initiator [xhr.js:177](#)

Usando API /pagarAhora

■ Network request

URL [:3200/pagarAhora](#)

Duration 1.64 s (320.94 ms network transfer + 1.32 s resource loading)

Request Method POST

Priority Alto

Mime Type application/json

Encoded Data 796 B

Decoded Body 527 B

Initiator [xhr.js:177](#)

Usando API para enviar correos

■ Network request

URL [api.emailjs.com/api/v1.0/email/send](#)

Duration 1.89 s (1.88 s network transfer + 5.38 ms resource loading)

Request Method POST

Priority Alto

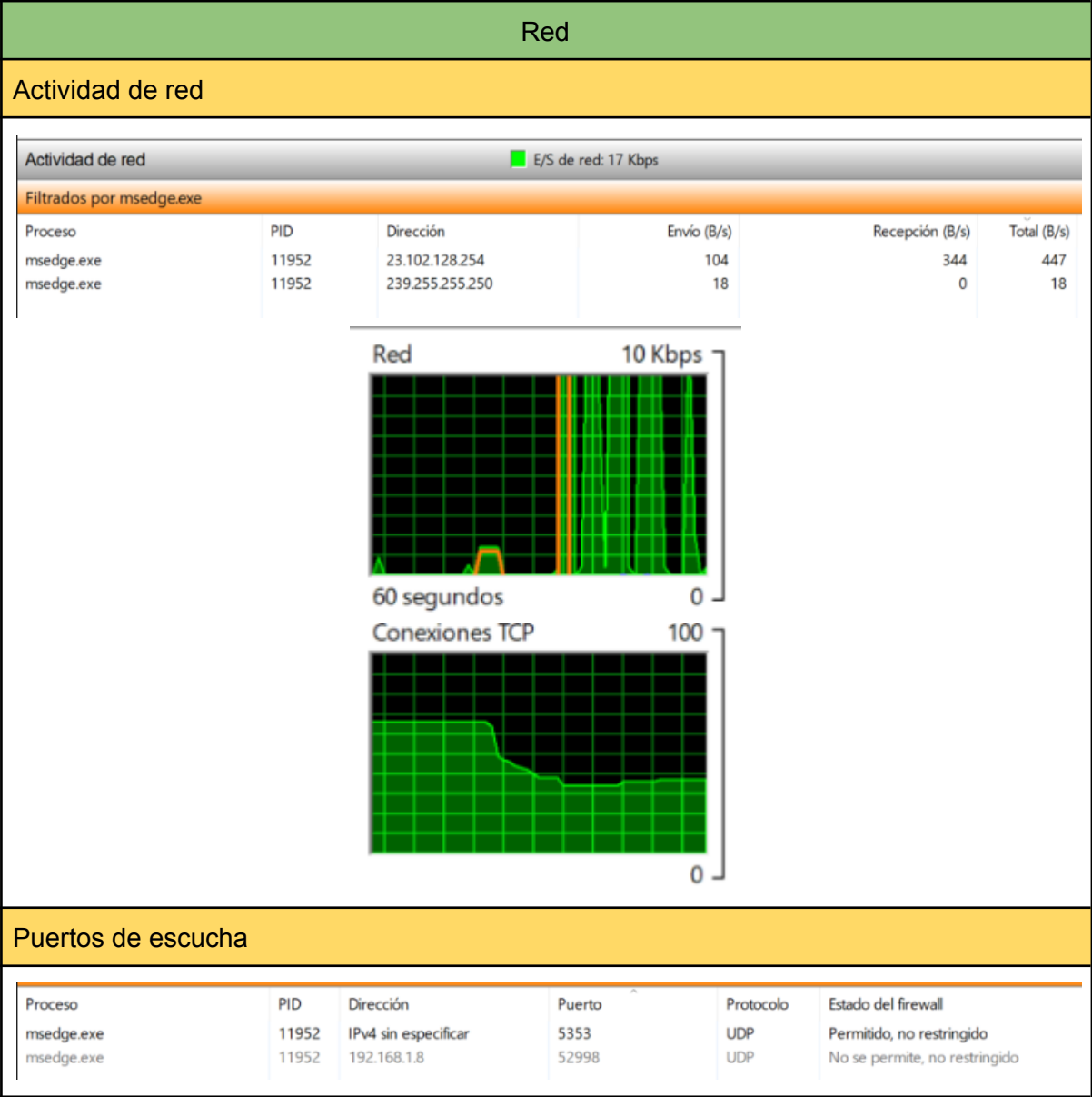
Mime Type text/plain

Encoded Data 206 B

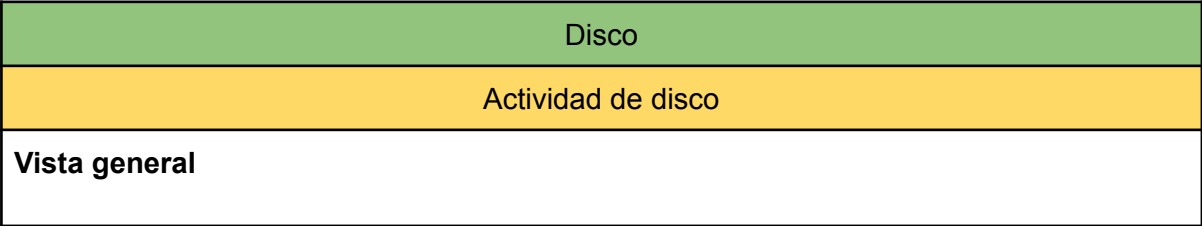
Decoded Body 2 B

Initiator [xhr.js:177](#)

- Esfuerzo



Puertos de escucha



Procesos con actividad de disco

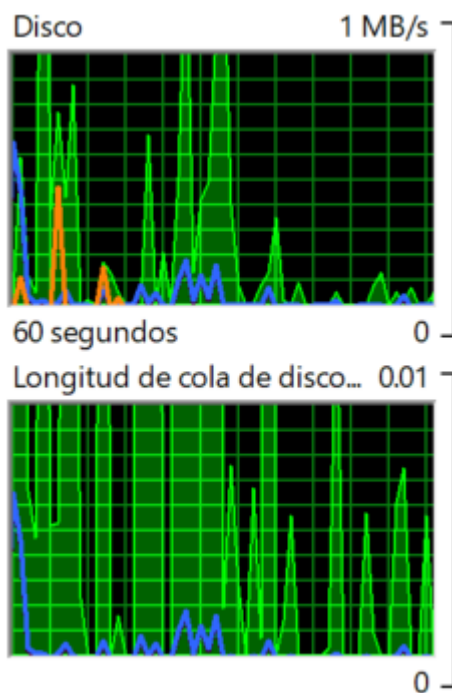
<input checked="" type="checkbox"/> Proceso	PID	Lectura (B/s)	Escritura (B/s)	Total (B/s)
<input checked="" type="checkbox"/> msedge.exe	11952	0	15,018	15,018

Vista detallada

Actividad de disco ■ E/S de disco: 0 KB/s ■ 2% de tiempo activo más alto

Filtrados por msedge.exe

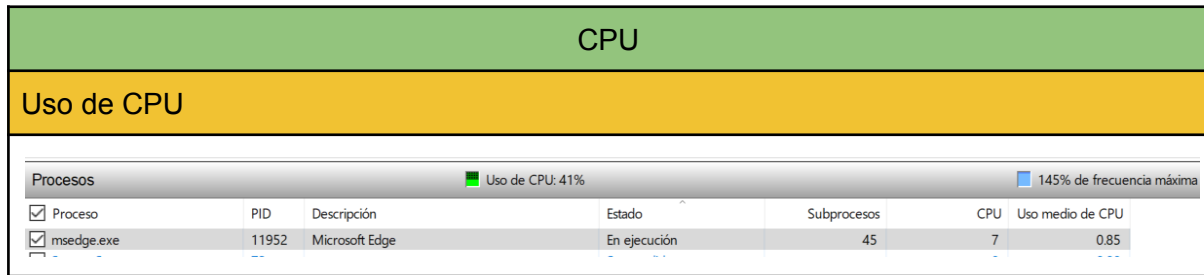
Proceso	PID	Archivo	Lectura (B/s)	Escritura (B/s)	Total (B/s)	Prioridad de E/S	Tiempo de respu...
msedge.exe	11952	C:\Users\efr...	0	682	682	Normal	326
msedge.exe	11952	C:\\$LogFile ...	0	1,646	1,646	Normal	20
msedge.exe	11952	C:\Users\efr...	0	413	413	Normal	4
msedge.exe	11952	C:\Users\efr...	0	9,879	9,879	Normal	2
msedge.exe	11952	C:\Users\efr...	0	658	658	Normal	2
msedge.exe	11952	C:\Users\efr...	0	407	407	Fondo	2
msedge.exe	11952	C:\Users\efr...	0	1,923	1,923	Fondo	1
msedge.exe	11952	C:\Users\efr...	0	857	857	Normal	0
msedge.exe	11952	C:\Users\efr...	0	194	194	Normal	0



Memoria

Memoria física utilizada

Procesos ■ 59% de memoria física usada						
<input checked="" type="checkbox"/> Proceso	PID	Errores graves/s	Asignación (KB)	Espacio de trabajo (KB)	Se puede compartir (KB)	Privada (KB)
<input checked="" type="checkbox"/> msedge.exe	11952	0	64,496	156,252	99,500	56,752



Uso de herramientas CI/CD



Jenkins

Jenkins es un servidor de automatización open source escrito en Java, ayuda en la automatización de parte del proceso de desarrollo de software mediante integración continua y facilita ciertos aspectos de la entrega continua. Admite herramientas de control de versiones como CVS, Subversion, Git, Mercurial, Perforce y Clearcase y puede ejecutar proyectos basados en Apache Ant y Apache Maven, así como secuencias de comandos de consola y programas por lotes de Windows.

Características



Integración continua y entrega continua

Como servidor de automatización extensible, Jenkins puede usarse como un servidor CI simple o convertirse en el centro de entrega continua para cualquier proyecto.



Fácil instalación

Jenkins es un programa autónomo basado en Java, listo para ejecutarse de inmediato, con paquetes para Windows, Linux, macOS y otros sistemas operativos similares a Unix.



Fácil configuración

Jenkins se puede instalar y configurar fácilmente a través de su interfaz web, que incluye comprobaciones de errores sobre la marcha y ayuda incorporada.



Complementos

Con cientos de complementos en el Centro de actualización, Jenkins se integra con prácticamente todas las herramientas en la cadena de herramientas de integración continua y entrega continua.



Extensible

Jenkins se puede extender a través de su arquitectura de complementos, lo que brinda posibilidades casi infinitas de lo que Jenkins puede hacer.



Repartido

Jenkins puede distribuir fácilmente el trabajo en varias máquinas, lo que ayuda a impulsar las compilaciones, las pruebas y las implementaciones en varias plataformas con mayor rapidez.

Utilización en este proyecto

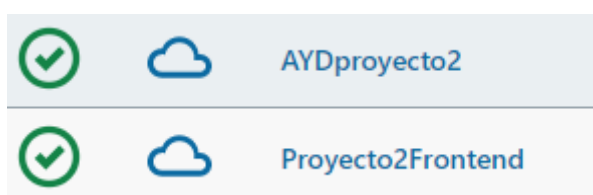
Administración de plugins

Adicionalmente a los plugins utilizados comúnmente, se instalaron los siguientes complementos.

- **NodeJS Plugin**
Proporciona integración de Jenkins para paquetes NodeJS y npm.
Versión: 1.4.0
- **Docker Pipeline**
Complemento de Jenkins que permite crear, probar y usar imágenes de Docker de proyectos de Jenkins Pipeline.
Versión: 1.26

Creación de tareas de tipo Pipeline

Para la automatización de integración y entrega de este proyecto, utilizamos dos tareas de tipo Pipeline, cada una de ellas monitorea los cambios agregados al repositorio donde se aloja la solución de nuestro proyecto.



La primera tarea, denominada AYDproyecto2, consulta cambios realizados al backend de nuestra aplicación continuamente, específicamente cada 1 minuto. Cuando se detecta un cambio, se ejecuta el siguiente script de Jenkins.

```
pipeline{
  agent any
  tools {nodejs "node"}
  stages{
    stage('git'){
      steps{
        git branch: 'feature/backend',
          url : 'https://github.com/erickace/AYD1_Proyecto2_7.git',
          credentialsId: '8d40f314-e3a0-4f08-a6e2-22f2b4582324'
      }
    }
    stage('install'){
      steps{
        sh 'cd /var/lib/jenkins/workspace/AYDproyecto2/server/
          && npm i
          && npm uninstall mocha
          && npm i mocha'
      }
    }
    stage('pruebas'){
      steps{
        sh '/var/lib/jenkins/workspace/AYDproyecto2/server/node_modules/.bin/mocha
          /var/lib/jenkins/workspace/AYDproyecto2/server/test/**'
      }
    }
    stage('docker-build'){
      steps{
        sh 'cd /var/lib/jenkins/workspace/AYDproyecto2/server/
          && docker build -t server .
          && docker stop server0
          && docker rm server0 -f'
      }
    }
    stage('docker-run'){
      steps{
        sh 'docker run --name "server0" -d -p 3200:3000 server'
      }
    }
    stage('end'){
      steps{
        echo 'end'
      }
    }
  }
}
```

Este script cuenta con 6 etapas, cada una se ejecuta después de la anterior.

- **Stage git**
Indicamos la rama del repositorio que se consultará, así mismo, indicamos las credenciales para acceder al mismo.
- **Stage install**
En esta etapa, se instalan todas las dependencias utilizadas para nuestro proyecto.
- **Stage pruebas**
En esta etapa realizamos las pruebas unitarias codificadas previamente, cada una de ellas debe devolver un resultado satisfactorio para completar la ejecución.

- **Stage docker-build**

Después de pasar la etapa de pruebas unitarias, se realiza la construcción de una imagen de Docker, esta contiene el resultado de los cambios de nuestro repositorio en la rama backend.

- **Stage docker-run**

Se genera un contenedor basado en la imagen generada en la etapa anterior, indicando el puerto que debe utilizar para su comunicación.

- **Stage end**

Indica que el proceso ha terminado.

La segunda tarea, denominada, Proyecto2Frontend, consulta la rama frontend de nuestro repositorio, en ella se aloja la construcción del sitio web de nuestra aplicación. Al igual que en la tarea anterior, la consulta al repositorio se realiza periódicamente, en lapsos de un minuto. Al detectar cambios, se ejecuta el siguiente script.

```
pipeline{
    agent any
    tools {nodejs "node"}
    stages{
        stage('git'){
            steps{
                git branch: 'feature/frontend',
                  url : 'https://github.com/erickace/AYD1_Proyecto2_7.git',
                  credentialsId: '8d40f314-e3a0-4f08-a6e2-22f2b4582324'
            }
        }
        stage('docker-build'){
            steps{
                sh 'cd /var/lib/jenkins/workspace/Proyecto2Frontend/design-layer/
                  && docker build -t serverweb .
                  && docker stop serverweb0
                  && docker rm serverweb0 -f'
            }
        }
        stage('docker-run'){
            steps{
                sh 'docker run --name "serverweb0" -d -p 3002:80 serverweb'
            }
        }
        stage('end'){
            steps{
                echo 'end'
            }
        }
    }
}
```

Las etapas de ejecución se describen a continuación.

- **Stage git**

Indicamos la rama del repositorio que se consultará, así mismo, indicamos las credenciales para acceder al mismo

- **Stage docker-build**

Se realiza la construcción de una imagen de Docker con la solución web de nuestra aplicación.

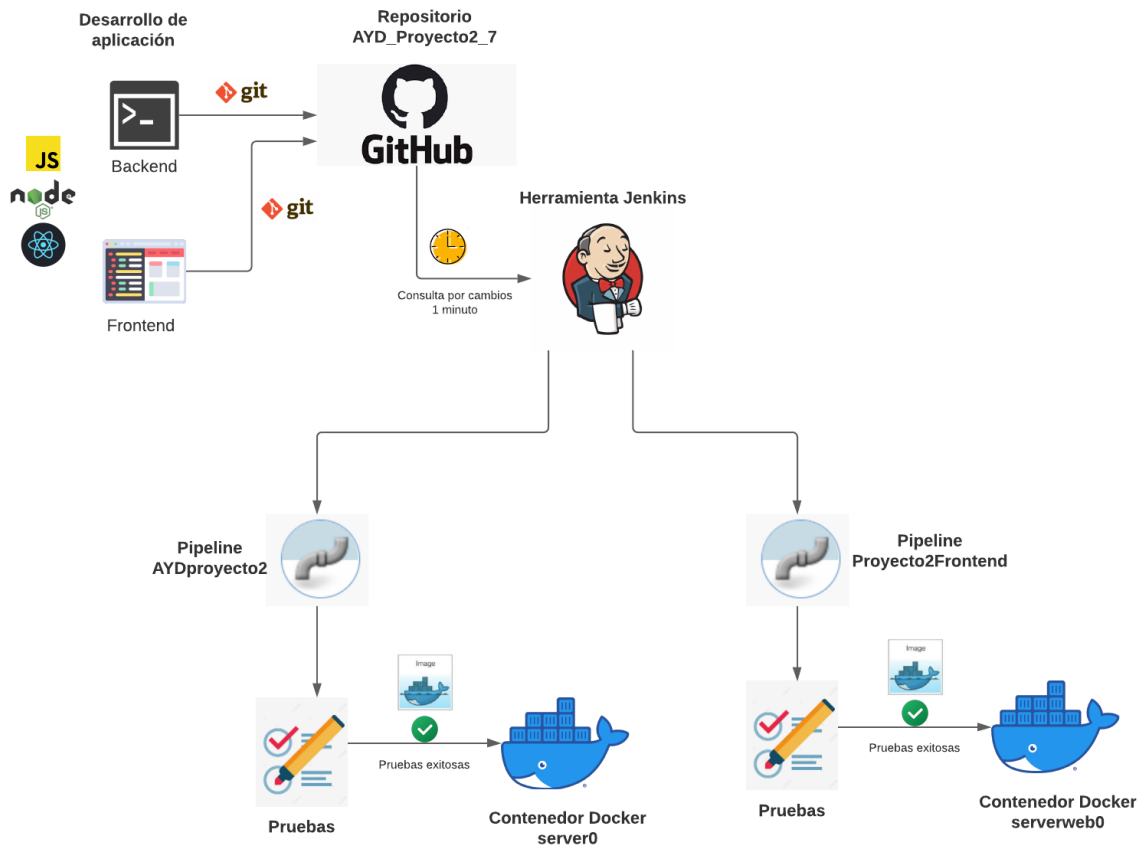
- **Stage docker-run**

Se genera un nuevo contenedor con la imagen generada anteriormente.

- **Stage end**

Indica que la ejecución terminó.

Diagrama de uso de Jenkins como herramienta de CI/CD



Jenkins

búsqueda

1

grupo7

Desconectar

Panel de Control

Nueva Tarea

Personas

Historial de trabajos

Relacion entre proyectos

Comprobar firma de archivos

Administrar Jenkins

Mis vistas

Lockable Resources

New View

Trabajos en la cola

No hay trabajos en la cola

Estado del ejecutor de construcciones

1 Inactivo

2 Inactivo

Todo

+

S	W	Nombre	Último Éxito	Último Fallo	Última Duración	
✓	☁	AYDproyecto2	12 Hor - #66	12 Hor - #64	19 Seg	🔗
✓	🔗	Proyecto2Frontend	1 Min 14 Seg - #7	8 Hor 21 Min - #3	1 Min 8 Seg	🔗

Icono: S M L

Guía de iconos

Atom feed para todos

Atom feed para fallos

Atom feed para los más recientes

añadir descripción

REST API

Jenkins 2.289.2

Back to Dashboard

Status

Changes

Construir ahora

Configurar

Borrar Pipeline

Full Stage View

GitHub

Rename

Pipeline Syntax

Git Log de consultas

Historia de tareas

Tendencia

find

x

66

2 Jul 2021 2:36

65

2 Jul 2021 2:33

64

2 Jul 2021 2:33

63

2 Jul 2021 2:19

Pipeline AYDproyecto2

Proyecto No. 2

Laboratorio de Análisis y Diseño de Sistemas 1

Sección N

Junio 2021

Grupo No. 7 - All for one

editar descripción

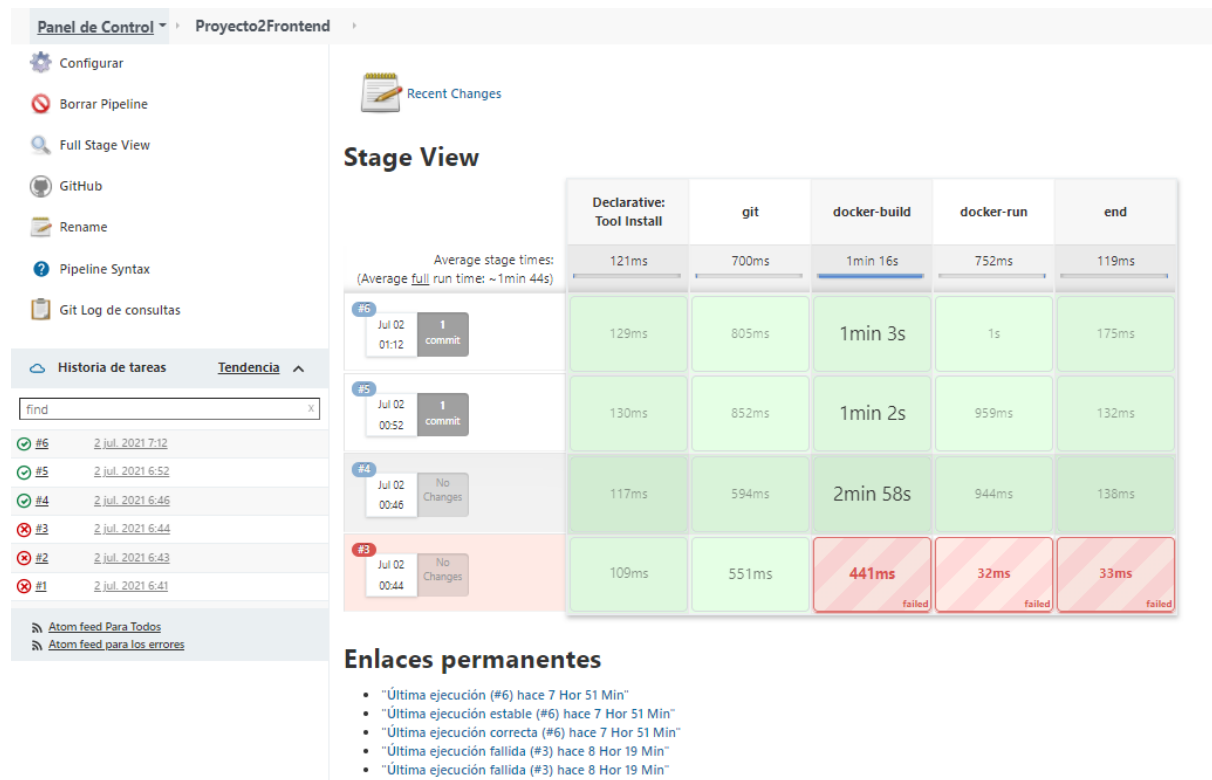
Desactivar el Proyecto

Recent Changes

Stage View

Average stage times:
(Average full run time: ~17s)

	Declarative: Tool Install	git	install	pruebas	docker-build	docker-run	end
	101ms	736ms	9s	1s	2s	350ms	170ms
66 Jul 01 2036 1 commit	102ms	775ms	10s	1s	5s	663ms	590ms
65 Jul 01 2033 No Changes	101ms	622ms	9s	1s	2s	692ms	228ms
64 Jul 01 2033 No Changes	84ms	575ms	9s	678ms failed	28ms failed	32ms failed	29ms failed



Plan de pruebas

- Descubrir defectos (para corregirlos)
 - Provocar fallas (forma de detectar defectos)
- Corrección de defectos
 - Hacer cambios en el sistema para remover los defectos.
- Comprobar que el sistema cumple con los requerimientos específicos (funcionales y no funcionales)
- Comprobar que el software hace lo que el usuario espera.
- Someter al sistema a grandes cargas o esfuerzos considerables (grandes volúmenes de datos en un corto periodo de tiempo).
- Tratar de burlar los controles de seguridad del sistema, para posteriormente hacer correcciones.
- Hacer pruebas para mostrar que el sistema cumple o no, con sus especificaciones de rendimiento.

Prueba	Tarea	Fecha inicio	Fecha Fin	Duración (min)
--------	-------	--------------	-----------	----------------

Unitaria	Login de usuario	29/06/2021	29/06/2021	4
	Registro de usuarios	29/06/2021	29/06/2021	5
	Obtener productos	29/06/2021	29/06/2021	3
	añadir al carrito	29/06/2021	29/06/2021	5
	Actualizar carrito	29/06/2021	29/06/2021	5
	Eliminar un producto del carrito	29/06/2021	29/06/2021	4
	Limpiar carrito	29/06/2021	29/06/2021	3
	ver carrito	29/06/2021	29/06/2021	3
	pagar ahora	29/06/2021	29/06/2021	5
	filtro por categoria	29/06/2021	29/06/2021	3

	ver categorías	29/06/2021	29/06/2021	3
	ver información de factura	29/06/2021	29/06/2021	3
Integración	Backend	29/06/2021	29/06/2021	10
	Frontend	29/06/2021	29/06/2021	10
Seguridad	Página de inicio	30/06/2021	30/06/2021	5
	Página productos	30/06/2021	30/06/2021	5
	Página carrito	30/06/2021	30/06/2021	5
Rendimiento	Obtener productos	30/06/2021	30/06/2021	4
	Categorías	30/06/2021	30/06/2021	4
	Login	30/06/2021	30/06/2021	4
	Obtener por categoría	30/06/2021	30/06/2021	4

	Agregar producto a carrito	30/06/2021	30/06/2021	5
	Leer carrito	30/06/2021	30/06/2021	3
	Actualizar carrito	30/06/2021	30/06/2021	4
	Pagar ahora	30/06/2021	30/06/2021	5
	Enviar correo	30/06/2021	30/06/2021	5
Validación	Validaciones en formularios.	30/06/2021	30/06/2021	15
Esfuerzo	Grandes volúmenes de datos en poco periodo de tiempo.	30/06/2021	30/06/2021	15

Detalle de pruebas unitarias

No.	Funcionalidad	Punto de acceso	Escenario	Estado esperado	Estado obtenido
1	Login de usuario	/login	Credenciales correctas	200	200
2	Login de usuario	/login	Correo electrónico incorrecto	500	500
3	Login de usuario	/login	Contraseña incorrecta	500	500
4	Login de usuario	/login	Credenciales incorrectas	500	
5	Registro de usuario	/registrarUsuario	Correo no ha sido registrado	200	200

6	Registro de usuario	/registrarUsuario	Correo ya ha sido registrado por otro usuario	500	500
7	Registro de usuario	/registrarUsuario	Faltan dirección del usuario	500	500
8	Registro de usuario	/registrarUsuario	Falta nombre de usuario	500	500
9	Obtener productos	/obtenerProductos	Obtener productos	200	200
10	Obtener un producto	/obtenerProducto/:idProducto	Producto existente	200	200
11	Obtener un producto	/obtenerProducto/:idProducto	Producto inexistente	200	200
12	Obtener un producto	/obtenerProducto/:idProducto	Se envia una cadena de caracteres como idProducto	200	200
13	Añadir al carrito	/agregarProductoCarrito	Producto no está en carrito - existencia suficiente	200	200
14	Añadir al carrito	/agregarProductoCarrito	Producto no está en carrito - existencia insuficiente	500	500
15	Añadir al carrito	/agregarProductoCarrito	Producto está en carrito - existencia suficiente	200	200
16	Añadir al carrito	/agregarProductoCarrito	Producto está en carrito - existencia insuficiente	500	500
17	Añadir al carrito	/agregarProductoCarrito	Producto inexistente	500	500
18	Actualizar carrito	/actualizarCarrito	Producto no esta en el carrito del cliente	500	500
19	Actualizar carrito	/actualizarCarrito	Agregar producto - existencia suficiente	200	200
20	Actualizar carrito	/actualizarCarrito	Agregar producto - existencia insuficiente	500	500
21	Actualizar carrito	/actualizarCarrito	Disminuir producto	200	200
22	Actualizar carrito	/actualizarCarrito	Cantidad igual a la existente en el carrito	200	200
23	Eliminar un producto del carrito	/eliminarProductoCarrito	Producto inexistente	500	500
24	Eliminar un producto del carrito	/eliminarProductoCarrito	Producto no esta en el carrito	500	500

25	Eliminar un producto del carrito	/eliminarProductoCarrito	Producto esta en el carrito	200	200
26	Limpiar carrito	/limpiarCarrito/:idUsuario	Usuario no tiene productos en carrito	200	200
27	Limpiar carrito	/limpiarCarrito/:idUsuario	Usuario tiene productos en carrito	200	200
28	Ver carrito	/leerCarrito	Usuario no tiene productos en carrito	200	200
29	Ver carrito	/leerCarrito	Usuario tiene productos en carrito	200	200
30	Pagar ahora	/pagarAhora	Usuario inexistente	500	500
31	Pagar ahora	/pagarAhora	No se especificó nit	500	500
32	Pagar ahora	/pagarAhora	No se especificó correo	500	500
33	Pagar ahora	/pagarAhora	No se especifico teléfono	500	500
34	Pagar ahora	/pagarAhora	Datos correctos	200	200
35	Filtro por categoria	/filtroCategoria/:idCategoria	Categoria correcta	200	200
36	Filtro por categoria	/filtroCategoria/:idCategoria	Categoría inexistente	500	200
37	Ver categorias	/categorias	Ver categorías	200	200
38	Ver información de factura	/factura/:idFactura	La factura indicada existe	200	200
39	Ver información de factura	/factura/:idFactura	La factura indicada no existe	500	500