



Manual de Usuario v1.0

Tabla de Contenidos

Tabla de Contenidos	1
Introducción	2
Descripción general	2
Acerca de Caché Sync	2
Propósito de Este Manual	2
Versión del Manual y de la Aplicación	2
Estructura del Manual	2
Requisitos del sistema	3
Requisitos del Cliente	3
Requisitos del Servidor	3
Instalación	4
Instalación del Backend (go)	4
Instalación del Frontend (React)	5
Palabras finales	5
Pantalla de Inicio	6
Elementos de Pantalla	6
Instrucciones de Uso	6
Confirmación de solicitud	7
Elementos de Pantalla	7
Instrucciones de Uso	7
Pantalla de Ejecución	8
Elementos de Pantalla	8
Instrucciones de Uso	9
Pantalla de Resumen y Estadísticas	10

Introducción

Descripción general

Bienvenido al Manual de Usuario de Caché Sync. Este manual está diseñado para proporcionar a los usuarios una guía completa sobre cómo utilizar Caché Sync, inicializando el servidor y cliente de manera efectiva. Ya sea que seas un usuario nuevo que está explorando las características por primera vez o un usuario experimentado que busca información específica, este manual tiene la información que necesitas.

Acerca de Caché Sync

Cache Sync es una poderosa herramienta diseñada para entender el correcto funcionamiento de los protocolos de coherencia de caché. Nuestra aplicación está dirigida a estudiantes avanzados de Ingeniería en Computadores específicamente aquellos que estén cursando el curso de Arquitectura de Computadores II y está diseñada para evaluar qué protocolo es más eficiente.

Propósito de Este Manual

El propósito de este manual es proporcionarte una guía detallada para aprovechar al máximo Caché Sync. Encontrarás instrucciones paso a paso, consejos útiles y respuestas a preguntas frecuentes.

Versión del Manual y de la Aplicación

Versión del Manual: **1.0.0** Versión de la Aplicación: **1.0.0**

Estructura del Manual

A lo largo de este manual, encontrarás información detallada sobre cómo utilizar cada función de Caché Sync, cómo descargar el repositorio de servidor y de cliente. Las secciones incluyen instrucciones de instalación, navegación, entre otros temas.

Te invitamos a explorar este manual y a utilizarlo como referencia cuando lo necesites. Caché Sync está diseñado para comprender de manera más eficiente conceptos de coherencia de caché, y este manual te ayudará a sacar el máximo provecho de todas sus características.

Requisitos del sistema

Para utilizar Caché Sync, asegúrate de cumplir con los siguientes requisitos del sistema tanto en el lado del cliente (front-end) como en el lado del servidor (back-end):

Requisitos del Cliente

- **Navegador Web Moderno:** Caché Sync es una aplicación web desarrollada en React, por lo que necesitas un navegador web moderno y actualizado. Se recomienda el uso de navegadores como Google Chrome, Mozilla Firefox, Safari o Microsoft Edge en sus versiones más recientes.
- **Conexión a Internet:** Para acceder a Caché Sync, debes estar conectado a Internet para cargar la aplicación y realizar solicitudes al servidor.
- **Habilitar JavaScript:** Asegúrate de que JavaScript esté habilitado en tu navegador. Caché Sync utiliza JavaScript para la funcionalidad interactiva.

Requisitos del Servidor

El backend de Caché Sync está desarrollado en Go y consta de varios componentes. Dentro de los puntos principales podemos abarcar:

- **Sistema Operativo:** El backend de Go es compatible con varios sistemas operativos, incluyendo Linux y Windows
- **Go (Golang):** Debes tener Go (Golang) instalado en el servidor. Puedes descargar Go desde el sitio web oficial de Go (<https://golang.org/>) e instalarlo siguiendo las instrucciones correspondientes a tu sistema operativo.
- **Bibliotecas y Dependencias:** El backend de Go puede requerir bibliotecas y dependencias adicionales para funcionar correctamente. Asegúrate de que todas las dependencias necesarias estén instaladas y configuradas según las instrucciones del proyecto.
- **Conectividad de Red:** El servidor debe tener conectividad de red para comunicarse con la aplicación front-end y para realizar operaciones de comunicación con otros componentes, como la cola (Queue) y la memoria principal (Main Memory).
- **Configuración y Despliegue:** Se debe asegurar de que el backend esté configurado correctamente para interactuar con los componentes específicos de Caché Sync, a nivel como el Program Generator, Queue, Cache, Interconnect, Main Memory y Communication. Configura las rutas y los puntos finales de API según sea necesario.

Instalación

A continuación, se describen los pasos para instalar y configurar tanto el backend como el frontend de Caché Sync. Asegúrate de seguir estos pasos cuidadosamente para garantizar un funcionamiento correcto.

Instalación del Backend (go)

1. Requisitos Previos:

- a. Asegúrate de tener Go (Golang) instalado en tu sistema. Puedes descargarlo desde el sitio web oficial de Go (<https://golang.org/>) e instalarlo siguiendo las instrucciones correspondientes a tu sistema operativo.

2. Clonar el Repositorio:

- a. Abre una terminal y ejecuta el siguiente comando para clonar el repositorio del backend desde GitHub:

```
git clone https://github.com/ce-itcr/CE4302-P1-2023-S2
```

3. Acceder al Directorio:

- a. Ingresa al directorio del repositorio clonado

```
cd CE4302-P1-2023-S2/src/Backend
```

4. Instalar Dependencias:

- a. Asegúrate de que tengas instalada la versión mínima requerida de Go, que es 1.13.0.
- b. Ejecuta los comandos necesarios para instalar las dependencias del proyecto.

```
go list -m all
```

5. Ejecutar el Programa:

- a. Inicia el backend ejecutando los comandos siguientes:

```
go run main.go
```

```
go build -o main
```

Instalación del Frontend (React)

1. Requisitos Previos:

- a. Asegúrate de tener Node.js instalado en tu sistema. Puedes descargarlo desde el sitio web oficial de Node.js (<https://nodejs.org/>) e instalarlo siguiendo las instrucciones correspondientes a tu sistema operativo.

2. Clonar el Repositorio:

- a. Abre una terminal y ejecuta el siguiente comando para clonar el repositorio del frontend desde GitHub (rama master):

```
git clone https://github.com/ce-itcr/CE4302-P1-2023-S2
```

3. Acceder al Directorio:

- a. Ingresa al directorio del repositorio clonado:

```
cd /CE4302-P1-2023-S2/src/Frontend
```

4. Instalar Dependencias:

- a. Ejecuta el siguiente comando para instalar todas las dependencias del proyecto:

```
npm install
```

5. Ejecutar la Aplicación:

- a. Una vez que se completen las instalaciones de dependencias, inicia la aplicación ejecutando:

```
npm run start
```

6. Se abrirá una pestaña de navegador web en localhost:3000, donde podrás acceder a Caché Sync

Palabras finales

Con estos pasos, habrás instalado tanto el backend como el frontend de Caché Sync en tu entorno de desarrollo. Asegúrate de que ambos componentes estén en funcionamiento antes de comenzar a utilizar la aplicación. Si encuentras algún problema durante la instalación, consulta la documentación específica del proyecto o comunícate con el equipo de desarrollo para obtener asistencia adicional.

Pantalla de Inicio

La pantalla de inicio es la primera pantalla que los usuarios encuentran al abrir Caché Sync. Su función principal es que los usuarios escojan cuál protocolo usar como ejecución.



Elementos de Pantalla

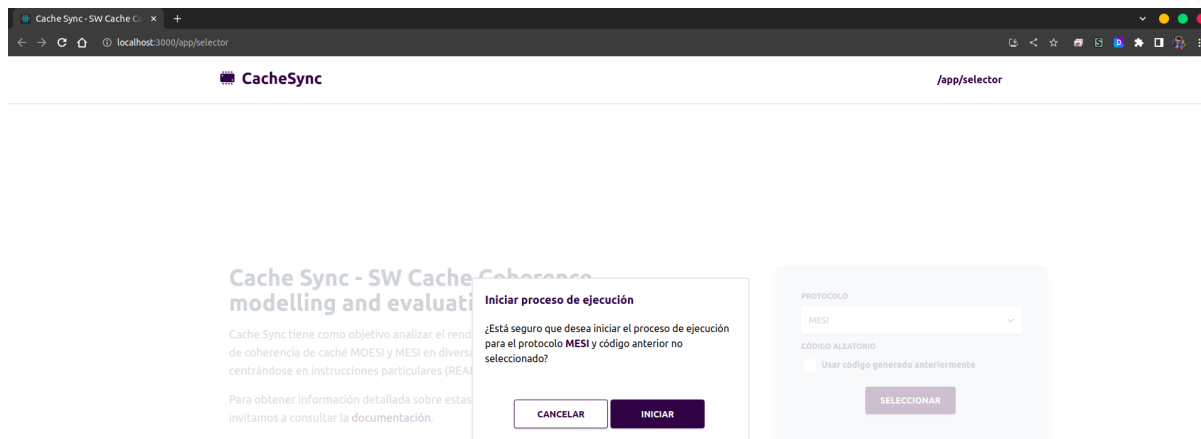
- **Sección descriptiva:** Describe el objetivo del sistema y tiene un link para que se pueda acceder a la documentación
- **Selección de protocolo:** Este representa un dropdown menu que tiene los protocolos a escoger. Los que tiene actualmente son MESI y MOESI.
- **Selección de código anteriormente generado:** Los usuarios pueden hacer clic a este checkbox para usar el código generado anteriormente

Instrucciones de Uso

1. En la parte de selección de modo tiene un menú desplegable con ambos modos, MESI y MOESI. Para seleccionar uno se debe dar click en la flecha de abajo en la interfaz de usuario y seleccionar el modo deseado.
2. En caso de ser la primera ejecución no se puede seleccionar la opción de "Usar código generado anteriormente".
 - a. Si ya hay al menos una ejecución realizada se puede seleccionar esta opción para ejecutar el código generado en la ejecución anterior y se puede hacer con el protocolo deseado. Esta opción sirve para comparar el desempeño de ambos protocolos realizados.
3. Seguidamente se puede dar click en SELECCIONAR para iniciar la ejecución del código.

Confirmación de solicitud

Esta pantalla tiene como objetivo principal confirmar la solicitud del usuario para dar inicio al proceso.



Elementos de Pantalla

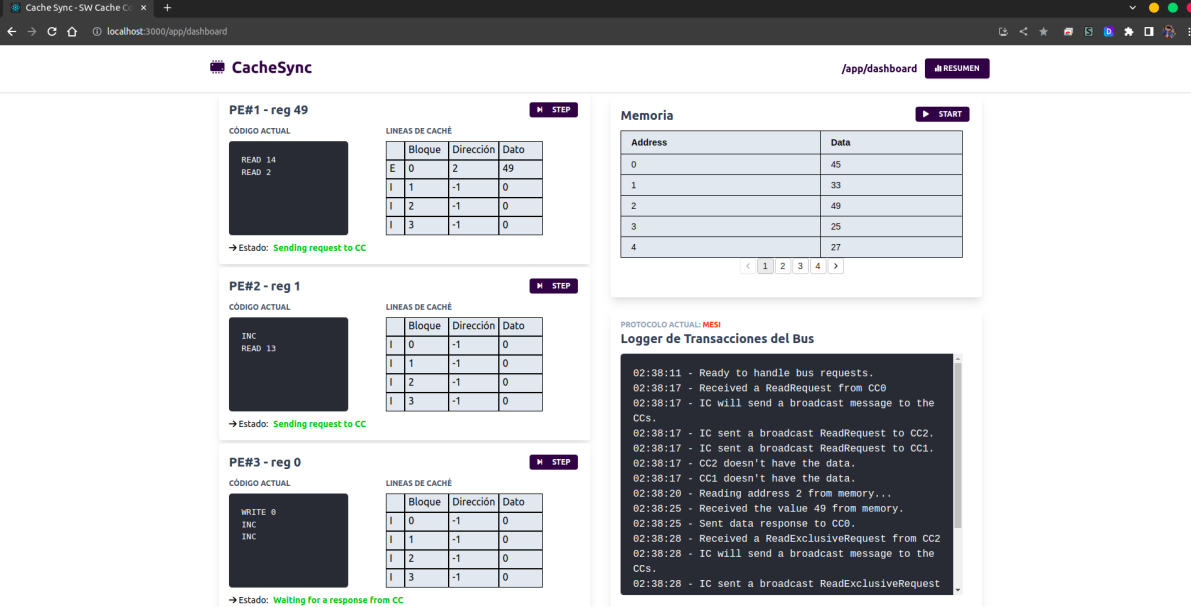
- **Botón de Confirmar:** Permite al usuario confirmar su decisión para comenzar el proceso con las opciones seleccionadas.
- **Botón de Cancelar:** Un principio heurístico es Libertad por parte del usuario, esto permite que el usuario pueda cancelar su decisión sobre iniciar con el proceso.

Instrucciones de Uso

1. En caso de seleccionar la opción de "Usar código generado anteriormente" se desplegará una ventana para confirmar si se desea mantener esta opción.
2. En caso de que se quiera continuar se hace click en CONFIRMAR, caso contrario en CANCELAR.

Pantalla de Ejecución

La pantalla de ejecución pretende reflejar el proceso a detalle de las acciones que se están ejecutando en los componentes de Interconnect, Memoria Principal, Caché para cada uno de los PE y el logger de transacciones.



The screenshot shows the CacheSync application interface. It features three panels for processing elements (PE#1, PE#2, PE#3) and a memory/log section.

PE#1 - reg 49 (M STEP button):

- CÓDIGO ACTUAL:** READ 14, READ 2
- LINEAS DE CACHE:**

	Bloque	Dirección	Dato
E	0	2	49
I	1	-1	0
I	2	-1	0
I	3	-1	0
- Estado:** Sending request to CC

PE#2 - reg 1 (M STEP button):

- CÓDIGO ACTUAL:** INC, READ 13
- LINEAS DE CACHE:**

	Bloque	Dirección	Dato
I	0	-1	0
I	1	-1	0
I	2	-1	0
I	3	-1	0
- Estado:** Sending request to CC

PE#3 - reg 0 (M STEP button):

- CÓDIGO ACTUAL:** WRITE 0, INC, INC
- LINEAS DE CACHE:**

	Bloque	Dirección	Dato
I	0	-1	0
I	1	-1	0
I	2	-1	0
I	3	-1	0
- Estado:** Waiting for a response from CC

Memoria (START button):

Address	Data
0	45
1	33
2	49
3	25
4	27

PROTOCOLO ACTUAL: MESI

Logger de Transacciones del Bus

```

02:38:11 - Ready to handle bus requests.
02:38:17 - Received a ReadRequest from CC0
02:38:17 - IC will send a broadcast message to the CCs.
02:38:17 - IC sent a broadcast ReadRequest to CC2.
02:38:17 - IC sent a broadcast ReadRequest to CC1.
02:38:17 - CC2 doesn't have the data.
02:38:17 - CC1 doesn't have the data.
02:38:20 - Reading address 2 from memory...
02:38:25 - Received the value 49 from memory.
02:38:25 - Sent data response to CC0.
02:38:28 - Received a ReadExclusiveRequest from CC2
02:38:28 - IC will send a broadcast message to the CCs.
02:38:28 - IC sent a broadcast ReadExclusiveRequest
  
```

Elementos de Pantalla

La pantalla está dividida visiblemente en dos columnas, donde en la columna izquierda se tiene el código generado para cada PE, el estado y las líneas de caché. Por otra parte, en la columna derecha se pueden observar todos los datos que se están guardando en memoria y en la parte inferior el logger de transacciones.

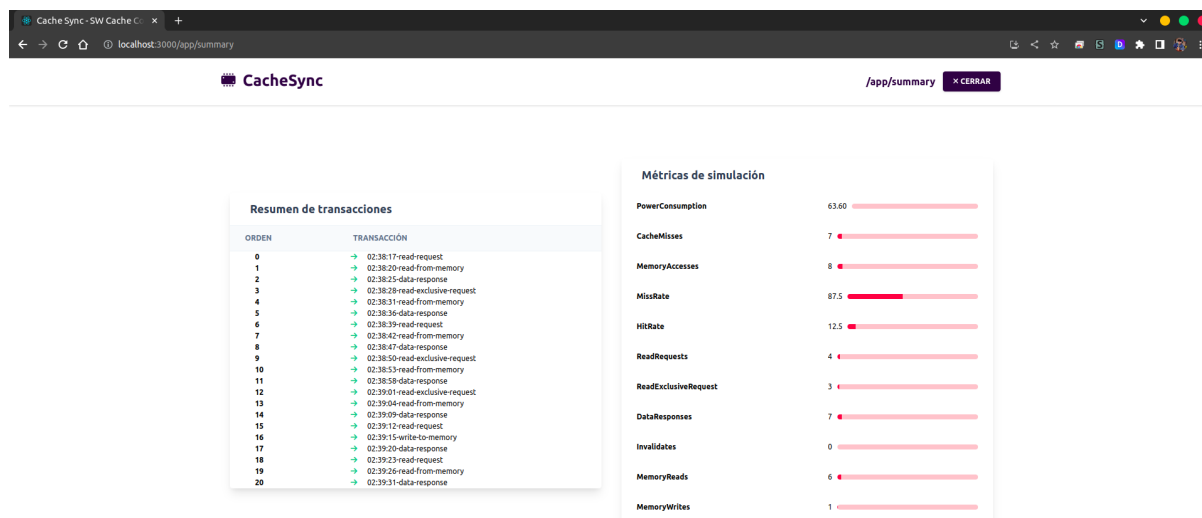
- **Botón de step:** Este botón lo que hace es ejecutar una única instrucción dentro del PE que se encuentra. Esta acción retornará cambios dentro de líneas de caché, estado, memoria y logger de transacciones.
- **Botón de start:** El botón de start permite ejecutar el programa de manera automática, donde se van a ejecutar las instrucciones de cada "pe" tomando diferentes aspectos de aleatoriedad de ejecución.
- **Botón de resumen:** Detiene la ejecución, se muestra un modal de confirmación y pasa a la siguiente pantalla donde se puede ver el resumen de transacciones.

Instrucciones de Uso

En esta pantalla se pueden notar varias partes importantes que explican secuencialmente lo que está pasando en ejecución.

- **Código generado:** En código generado se puede observar el código faltante a ejecutar en este processing element. En este se pueden ver 3 operaciones posibles, WRITE, READ o INC. WRITE y READ tienen una dirección de memoria asociada. El WRITE escribe el dato actual en el registro del processing element a la dirección de memoria asignada, el READ lee del caché o memoria el dato de memoria más actualizado del dato. El INC aumenta el registro del processing element.
- **Estado:** En estado se puede ver el estado actual del processing element. En este se puede ver si está ejecutando código, esperando por una respuesta o finalizado.
- **Líneas de Caché:** En esta parte se encuentran los datos y su estado actual. Al iniciar la ejecución este no cuenta con ningún dato pero ni estado pero al cargar o escribir datos este se modifica. El dato está en binario y los estados posibles dependen del protocolo seleccionado. En MESI se pueden observar estas siglas, que significan Modified, Exclusive, Shared y Invalid. En MOESI se pueden observar los estados Modified, Owner, Exclusive, Shared y Invalid.
- **Memoria Principal:** Esta es la representación de la memoria RAM. En esta se pueden ver todos los datos, pero estos datos no son los datos más actualizados.
- **Logger de Transacciones:** En esta parte se pueden observar cada acción que ocurren en todos los processing element, caché, memoria principal y estados de caché.
- **Step:** Con este botón se avanza una instrucción en el processing element asignado.
- **Start:** Este botón comienza una ejecución de todos los processing elements.

Pantalla de Resumen y Estadísticas



En esta pantalla se pueden observar algunas métricas de la ejecución actual así como el resumen de las transacciones y el orden en los que son ejecutados. Entre estas métricas se observan:

- **Miss Rate:** Se refiere a la proporción de veces que el programa encontró un dato válido en caché.
- **Hit Rate:** Se refiere a la proporción de veces que el programa no encontró un dato válido en caché.
- **Caché Hits:** Se refiere al número de veces que hubo hits .
- **Caché Miss:** Se refiere al número de veces que hubo misses.
- **Power Consumption:** Se refiere a una potencia potencial que podría ser consumida por el bus por cada transacción realizada en este. Es importante mencionar que esto es un peso potencial que podría tener y no datos garantizados.
- **Memory Accesses:** Se refiere a la cantidad de veces que se tuvo que acceder a memoria por cache misses.
- **Cantidad por instrucción:** Muestra la cantidad de instrucciones de tipo Write, Read, Increase, Invalidate y Data Responses que se realizaron en ejecución del código.