UNIVERSIDAD TÉCNICA DEL NORTE


FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA:

# ESTUDIO DE LAS ARQUITECTURAS TECNOLÓGICAS (APPSHELL, RESPONSIVE DESIGN, SOA) EN LA CONSTRUCCIÓN DE WEB PROGRESIVA PARA FORTALECER LA GESTIÓN DE PEDIDOS DE LOS LOCALES MIPYMES DE TIPO CAFÉ-RESTAURANT EN LA CIUDAD DE OTAVALO

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN SISTEMAS COMPUTACIONALES

AUTORA:

Andrea Carina Bautista Peñaquishpe

DIRECTOR:

Ing. Vicente Alexander Guevara Vega MSc.

Ibarra, 2019

**UNIVERSIDAD TÉCNICA DEL NORTE**

**BIBLIOTECA UNIVERSITARIA**

**AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

## 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

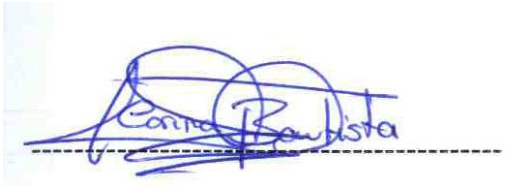
| **DATOS DE CONTACTO** | |
|---|---|
| **CÉDULA DE IDENTIDAD:** | 1004386643 |
| **APELLIDOS Y NOMBRES:** | BAUTISTA PEÑAQUISHPE ANDREA CARINA |
| **DIRECCIÓN:** | OTAVALO-IMBABUELA BAJO |
| **EMAIL:** | acbautistap@utn.edu.ec |
| **TELÉFONO MÓVIL:** | 0994721258 |

| **DATOS DE LA OBRA** | |
|---|---|
| **TÍTULO:** | ESTUDIO DE LAS ARQUITECTURAS TECNOLÓGICAS (APPSHELL, RESPONSIVE DESIGN, SOA) EN LA CONSTRUCCIÓN DE WEB PROGRESIVA PARA FORTALECER LA GESTIÓN DE PEDIDOS DE LOS LOCALES MIPYMES DE TIPO CAFÉ-RESTAURANT EN LA CIUDAD DE OTAVALO. |
| **AUTOR (ES):** | BAUTISTA PEÑAQUISHPE ANDREA CARINA |
| **FECHA: AAAAMMDD** | |
| SOLO PARA TRABAJOS DE GRADO | |
| **PROGRAMA:** | ☐ PREGRADO ☐ POSTGRADO |
| **TITULO POR EL QUE OPTA:** | INGENIERA EN SISTEMAS COMPUTACIONALES |
| ASESOR /DIRECTOR: | ING. VICENTE ALEXANDER GUEVARA VEGA MSC. |

## 2. CONSTANCIAS

La autora manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es la titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 15 días del mes de mayo de 2019



Andrea Carina Bautista Peñaquishpe

c.c.: 1004386643

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA

# CERTIFICACIÓN

En calidad de Director del trabajo de grado titulado: **ESTUDIO DE LAS ARQUITECTURAS TECNOLÓGICAS (APPSHELL, RESPONSIVE DESIGN, SOA) EN LA CONSTRUCCIÓN DE WEB PROGRESIVA PARA FORTALECER LA GESTIÓN DE PEDIDOS DE LOS LOCALES MIPYMES DE TIPO CAFÉ-RESTAURANT EN LA CIUDAD DE OTAVALO.** certifico que el presente trabajo fue desarrollado por la señorita Andrea Carina Bautista Peñaquishpe, bajo mi supervisión.

MSc. Alexander Guevara

DIRECTOR

## DEDICATORIA

Este trabajo está dedicado a Dios, a mi mamá: Anita Peñaquishpe, a mis hermanas y familia, quienes me apoyaron incondicionalmente para alcanzar el objetivo de culminar mis estudios.

## AGRADECIMIENTO

Agradezco a Dios por la vida, por todas las bendiciones y fortalezas que me otorgo para salir adelante en este arduo camino de estudiar una ingeniería, que resultó un poco difícil, pero no imposible de culminar. También a mi Mamá: Anita Peñaquishpe siendo padre y madre, quien a través del tiempo fuer el pilar más importante en mi vida en todo lugar y en todo momento, de ella aprendí que la constancia es una de las claves del éxito y que las cosas se deben ganar con esfuerzo y perseverancia, siempre estaré muy agradecida con ella por su apoyo incondicional.

A mis hermanas: Erika Collahuazo y Paulina Collahuazo; y a mi novio: Danny De la Cruz, quienes siempre estuvieron a mi lado con ese positivismo que se necesita para que las cosas resulten de la mejor manera.

A mis docentes de aula, quienes compartieron sus enseñanzas lo largo de esta vida universitaria de manera correcta y amigable, como si fuéramos compañeros, de ellos me llevo una grata experiencia de aprendizaje.

Finalmente, a mi Director de Trabajo de Grado, al MSc. Alexander Guevara, quien me brindó su apoyo al ser mi tutor y compartir gran parte de su sabiduría, siendo un pilar importante en todo el proceso de investigación, su experiencia y capacidades me ayudaron a culminar con éxito el presente proyecto.

## Tabla de Contenido

## Índice de Figuras

## Índice de Cuadros

## Resumen

El presente trabajo de titulación está basado en el estudio de tres arquitecturas para la construcción de aplicaciones web progresivas, como Responsive Design, SOA y App Shell.

Para el estudio, se aplicó el estándar ISO 42010, que es un estándar basado en la descripción de una arquitectura con el propósito de proporcionar un análisis y soporte mediante el uso de las descripciones de la arquitectura. Sin embargo, para aplicar el Estándar es necesario ser guiado a lo largo del ciclo de vida de una arquitectura, para ello está guiado por el Libro de Arquitectura de Software de Humberto Cervantes Maceda.

El Capítulo Uno detalla los conceptos sobre la ingeniería web y la evolución de las arquitecturas, incluida la descripción de las tres arquitecturas para el estudio.

El Capítulo Dos detalla lo esencial de esta tesis, que es una investigación basada en el ciclo de vida de una Arquitectura (Requisitos, Diseño, Documentación, Evaluación e Implementación), a lo largo del proceso de investigación y a través de pequeños laboratorios aplicados, cada arquitectura se midió con tres indicadores principales, tales como: como Funcionalidad, Adaptabilidad y Rendimiento, logrando la arquitectura ideal "App Shell" para la construcción de Web Progresivas de tipo informativo: cumple con características esenciales como: trabajar sin conexión, adaptarse a cualquier dispositivo, entre otras. Sin embargo, se espera que en el futuro una Web progresiva transaccional pueda tener un pequeño motor de base de datos para trabajar sin conexión.

## Abstract

In the present titling work, it is based on the study of three architectures for the construction of progressive web applications, such as Responsive Design, SOA and App Shell.
For the study, the ISO 42010 standard was applied, which is a standard based on the description of an architecture with the purpose of providing an analysis and support through the use of the descriptions of the architecture. However, to apply the Standard it is necessary to be guided throughout the life cycle of an architecture, for this it is guided by the Software Architecture Book of Humberto Cervantes Maceda.

Chapter One details the concepts of web engineering and the evolution of architectures, including the description of the three architectures for the study.
Chapter Two details the essence of this thesis, which is a research based on the life cycle of an Architecture (Requirements, Design, Documentation, Evaluation and Implementation), throughout the research process and through small applied laboratories, each Architecture was measured with three main indicators, such as: Functionality, Adaptability and Performance, achieving the ideal architecture "App Shell" for the construction of Web Progressives of informative type: meets essential features such as: work offline, adapt to any device , among other. However, it is expected that in the future a progressive transactional Web may have a small database engine to work offline.

## INTRODUCCIÓN

## PROBLEMA

## ANTECEDENTES

La Web ha sufrido grandes cambios, que han obligado a utilizar herramientas y técnicas basadas en la Ingeniería de Software para poder garantizar el buen funcionamiento y administración de los sitios Web. La Ingeniería de Software comprende métodos, estándares, metodologías y buenas prácticas que son utilizadas para el desarrollo y mantenimiento de un Software.

Para garantizar el buen funcionamiento y mantenimiento es necesario que el Software sea de calidad, para ello debe cumplir con atributos como: usabilidad, navegabilidad, seguridad, mantenibilidad, entre otros, que hace posible por un lado la eficiencia del artefacto Web y por ende la satisfacción del usuario final. Por lo tanto, la Ingeniería Web está basada en la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de las aplicaciones de alta calidad.

Dentro de la Ingeniería Web se encuentra inmersa las aplicaciones móviles que viene a ser un pequeño programa que se puede descargar de la Web (en muchos casos de tiendas de aplicaciones on-line) para ser instalado en el teléfono Smartphone, tableta o reproductor MP3 de forma nativa, a través de una conexión a internet.

## SITUACIÓN ACTUAL

La mayoría de desarrolladores de aplicaciones de hoy en día, construyen Aplicaciones Nativas para dispositivos móviles o en un entorno Web, cada una de ellas presenta ciertas ventajas y desventajas, así como la tecnología que está implicada en su desarrollo.
Las Aplicaciones Nativas, son exclusivamente desarrolladas para un determinado sistema operativo, llamado Software Development Kit1 o SDK, como por ejemplo las plataformas como: Android, iOS de Apple o Windows Phone, pero resultan ser costosas porque es necesario crear aplicaciones para cada sistema operativo móvil (Alarcon, 2016).

[1] SDK: Kit de desarrollo de software

Por lo que respecta las Aplicaciones de entorno Web, son aquellas a las que se accede a través de una URL[2] utilizando un interpretador como un navegador Web y que se adaptan a cualquier dispositivo utilizando librerías de tipo responsivo, pero su acceso es limitado: la necesidad de conexión permanente a Internet hacen que el acceso a estas aplicaciones no se encuentren al alcance de todos los usuarios (Alarcon, 2016).

**PROSPECTIVA**

Para solucionar aquellos inconvenientes, nace un nuevo paradigma llámese un punto de disrupción entre las Aplicaciones Nativas para dispositivos móviles y Aplicaciones de entorno Web, las llamadas Aplicaciones Web Progresivas en inglés “Progressive Web Apps” para aprovechar las nuevas características compatibles con navegadores modernos que ofrecerá un mejor rendimiento, adaptabilidad, usabilidad y funcionalidad, en cualquier dispositivo, manejando así el término BYOD traducido al español “ Trae tu propio dispositivo” adaptado por empresas como Microsoft, permitiendo a los usuarios utilizar sus propios dispositivos (ordenadores portátiles, Smartphone y tabletas) para acceder a cualquier aplicación.

De hecho, si empresas adoptan este tipo de aplicaciones, la información proyectada no solo sería visible para usuarios que tenga conectividad a internet de forma online, sino a varios usuarios con conexión offline.

Con el estudio a realizarse permitirá realizar la documentación necesaria para que estudiantes y profesionales interesados en la construcción de Aplicaciones Web Progresivas puedan incrementar sus conocimientos sobre esta temática.

**PLANTEAMIENTO DEL PROBLEMA**

La falta de bibliografía y un alto grado de desconocimiento acerca de las Aplicaciones Web Progresivas influye a que los desarrolladores se inclinen por realizar Aplicaciones en entornos Web o de forma nativa para dispositivos móviles, dejando de lado las ventajas que ofrece las Aplicaciones Web Progresivas como se muestra en la Figura 1.

[2] URL: Sus siglas en inglés significa "Uniform Resource Locator", en español, "Localizador Uniforme de Recursos**".**

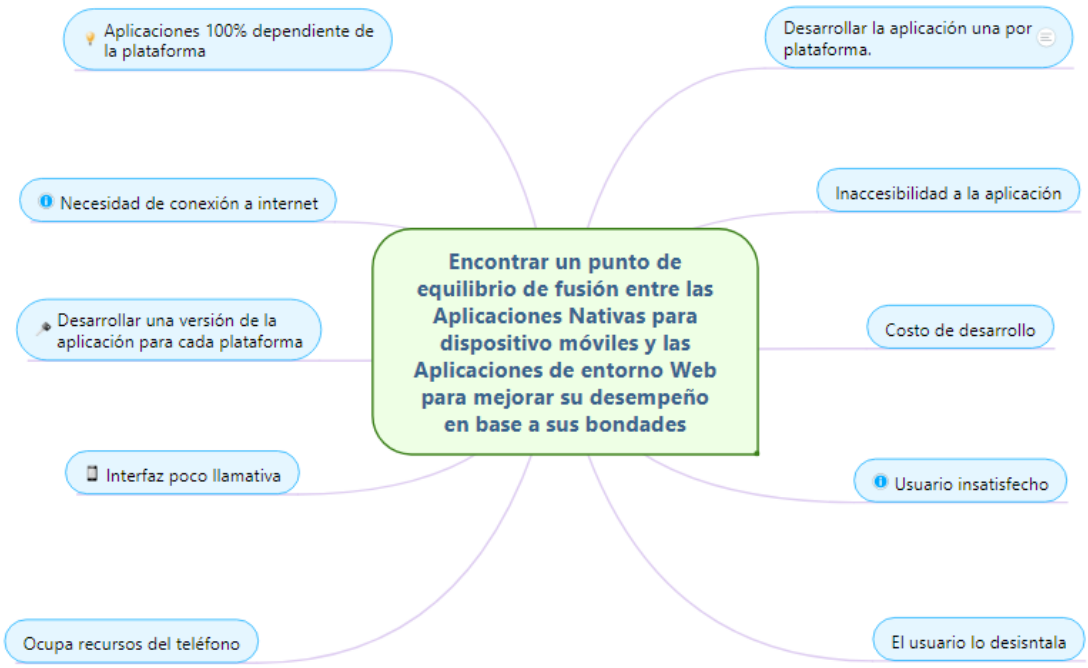



***Figura:*** *Árbol de Problemas*

***Fuente****: Propia*

Analizando estos aspectos, se realiza la siguiente interrogante:

*¿Cómo encontrar un punto de equilibrio de fusión entre las Aplicaciones Nativas para dispositivos móviles y las Aplicaciones de entorno Web para mejorar su desempeño en base a sus bondades?*

## OBJETIVOS

### OBJETIVO GENERAL

Estudiar las Arquitecturas Tecnológicas (App Shell, Responsive Design, SOA) en base a la Norma ISO 42010, en la construcción de Web Progresiva para fortalecer la gestión de pedidos de los locales Mi Pymes de tipo Café-Restaurant en la ciudad de Otavalo.

### OBJETIVOS ESPECÍFICOS

- Generar una caracterización de los entornos de Aplicaciones de entorno Web y Aplicaciones Nativas para dispositivos móviles.
- Generar un estudio comparativo de las arquitecturas tecnológicas App Shell, Responsive Design, SOA en base a la Norma ISO 42010, en la construcción de Aplicaciones Web Progresivas basado en indicadores de performance, adaptabilidad y funcionalidad.

- Implementar la arquitectura estudiada para la construcción de una Aplicación Web Progresiva para fortalecer el proceso de gestión de pedidos del Café- Restaurant “Deli- Café Restaurant” de la ciudad de Otavalo.
- Validar los resultados mediante la teoría de Chi cuadrado (X2).

**ALCANCE**

El presente proyecto tiene como finalidad estudiar las Arquitecturas Tecnológicas para mejorar la construcción de Web Progresiva.

Módulos del Aplicativo web:

- Módulo de autenticación

  Este módulo tiene como funcionalidad principal limitar a usuario externos no registrados a las funcionalidades que tiene el sistema según el rol asignado previamente al usuario.

- Módulo para registro

  Este módulo será el encargado de registrar la información de los diferentes platos del Menú para guardarlas en una base de datos y mostrar en la carta digital.

- Módulo gestión de pedidos

  Este módulo será el encargado de registrar el pedido que realice el cliente de la carta digital.

**JUSTIFICACIÓN**

El proyecto se lo realizará, por la necesidad de mostrar que arquitectura tecnológica ayudará a la construcción de Aplicaciones Web Progresivas para su desarrollo.

Los beneficiarios directos de esta herramienta será el gerente del Restaurante que haga uso del aplicativo web y los beneficiarios indirectos serán los clientes que utilicen el aplicativo Web progresivo para realizar el pedido de la carta digital. Por otra parte, en el presente estudio creará la documentación en español de las Aplicaciones Web Progresivas como un aporte para estudiantes y profesionales en el desarrollo y conocimiento de la funcionalidad y de la arquitectura de una Aplicación Web Progresiva.

# CAPÍTULO I

## Revisión bibliográfica

### 1.1. Ingeniería de Software

El software de un computador desde un inicio ha sido beneficioso para quienes los utilizan porque en ese transcurso satisfacen las necesidades sin fallos durante largos periodos, y hace que su modificación sea más fácil.

El software de computadora es el producto que construyen los programadores y al que después le dan mantenimiento durante un largo tiempo (Roger S. Pressman, 2010, pág. 1).

En la actualidad, el software sigue siendo un papel muy importante porque es considerado como un producto que transforma la información generada de un sin número de fuentes.

(Roger S. Pressman, 2010) Refiere lo siguiente. La ingeniería de Software comprende todos los aspectos que conllevan a la producción del software desde la etapa de herramientas hasta el compromiso de la calidad, como se muestra en la Figura 1.

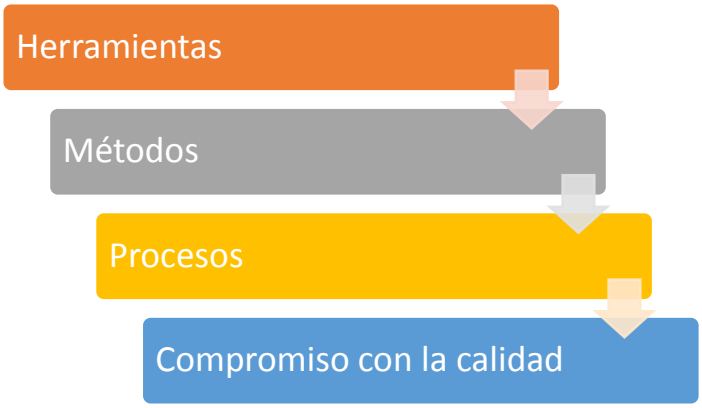



**Fig 1:** Capas de la Ingeniería de Software
**Fuente:** Propia

El principal elemento es la *capa proceso*, porque establece una estructura para que los desarrolladores[3] busquen y elijan las mejores acciones y tareas de manera oportuna para construir un software de calidad de acuerdo a las necesidades de quienes la usaran. (Roger S. Pressman, 2010).

[3] Desarrollador: es un programador que se dedica al proceso de desarrollo de software y que a su vez contribuyen a una visión general de un proyecto.

Los métodos aportan tareas como: análisis de requerimientos, diseño, desarrollo y pruebas para la elaboración del software (Roger S. Pressman, 2010).

Las herramientas proporcionan un apoyo a los métodos y los procesos para un mejor desarrollo de software (Roger S. Pressman, 2010).

Por lo tanto, la Ingeniería de Software comprende procesos, métodos y herramientas que se utilizan para asegurar la calidad del software, asimismo ayuda a resolver problemas definidos por un conjunto de principios fundamentales.

### 1.1.1. Ingeniería Web

La ingeniería Web[4] es un área que abarca procesos, técnicas y modelos orientados a los entornos Web. Consiste en la aplicación de metodologías sistemáticas en el desarrollo eficiente y evolución de aplicaciones de alta calidad en la World Wide Web[5] (Jarquín, 2014).

Por consiguiente, la ingeniería Web surgió como una nueva disciplina orientada a solucionar todos aquellos problemas que implicaba una baja calidad en el software, es decir, que existía una carencia en el proceso de construcción o desarrollo.

La calidad se refiere a las características que se pueden comparar con estándares como longitud, color, entre otros. A nivel de software existen unas métricas técnicas que proporcionan una manera sistemática de valorar claramente la calidad que son: (Jarquín, 2014):

**a) Usabilidad**

La definición formal que hace referencia la norma ISO 9241-11, indica que es “la medida en la que un producto se puede usar por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado” (Chipantiza & Olivo, 2015). Además, la usabilidad hace referencia a que el producto para ser fácil de usar por las personas, debe comprender tres elementos importantes como se muestra en la Figura 2.

[4] Web: Conjunto de información que se encuentra en una dirección determinada de internet.

[5] World Wide Web(WWW): es un conjunto de protocolos que permite la consulta remota de archivos de hipertexto.

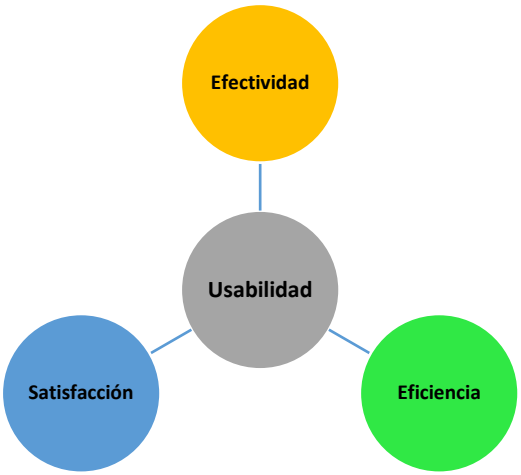



**Fig 2:** Elementos de la Usabilidad
**Fuente:** Propia

Los beneficios de la usabilidad en el desarrollo de software están siempre ligados tanto para los clientes como para los desarrolladores.

**b) Funcionalidad**

Consiste en la evaluación de las características y capacidades del programa, generalidades de las funcionalidades y la seguridad del software; que son primordiales para satisfacer las necesidades explicitas e implícitas cuando el software es utilizado bajo ciertas condiciones o políticas dentro de la organización.

**c) Fiabilidad**

Consiste en medir la frecuencia y gravedad de los fallos, la exactitud de las salidas ante una petición del usuario (respuestas) del sistema, tiempo medio de fallos y la capacidad que tenga en la recuperación de un posible fallo.

**d) Portabilidad**

Capacidad del producto software para ser migrado de un entorno a otro.

**e) Eficiencia**

Consiste en la capacidad del software para proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados bajo, la velocidad de procesamiento, y tiempo de respuesta.

**f) Mantenibilidad**

Capacidad del producto software para ser modificado. Las modificaciones podrían incluir correcciones, mejoras o adaptación del software a cambios en el entorno, y requisitos y especificaciones funcionales

Además dentro del proceso de ingeniería Web existe actividades que forman parte del marco de trabajo que serán aplicables a cualquier aplicación Web, independientemente del tamaño y complejidad de la misma (Roger S. Pressman, 2010) que se detallan a continuación:

- Comunicación con el cliente

  La comunicación con el cliente se caracteriza por medio de dos grandes tareas: el análisis del negocio que define el contexto empresarial-organizativo y la formulación es una actividad de recopilación de requisitos que involucran a todos los participantes.
- Planeación

  Se crea el plan del proyecto para el incremento de las aplicaciones Web. El plan consiste en una definición de tareas y un calendario de plazos respecto al período establecido para el desarrollo del proyecto.
- Modelado

  Las labores convencionales de análisis diseño de la ingeniería del software se adaptan al desarrollo de las aplicaciones Web se mezclan y luego se funden en una actividad de modelado de la ingeniería Web.
- Construcción

  Las herramientas y la tecnología de la ingeniería Web se aplican para construir las aplicaciones Web que se ha modelado.
- Despliegue

  Las aplicaciones Web se configuran para su ambiente operativo, se entrega a los usuarios finales y luego comienza un período de evaluación. La retroalimentación acerca de la evaluación para realizar los procesos respectivos.

Por ende, para garantizar el buen funcionamiento y mantenimiento de los sitios web, la ingeniería Web cumple un papel muy importante, para alcanzar el éxito en cualquier organización, y todo aquello que se pueda considerar como servicio; puesto que está inmersa la calidad que hace posible la eficiencia del artefacto web y la satisfacción del usuario final.

### 1.1.2. Aplicaciones Nativas

Las aplicaciones también conocidas como apps han estado presentes en los teléfonos desde hace tiempo; de hecho estaban incluidas en los sistemas operativos de Nokia y BlackBerry (Vittone, 2017). Además, están evolucionando hacia ambientes de cómputos sofisticados que no solo proveen características como funciones de cómputo y contenido para el usuario final, sino que también están integradas con bases de datos corporativas y aplicaciones de negocios.

Por tanto, una aplicación no deja de ser un software que es exclusivamente para móviles, así como los programas son para las computadoras y laptops. Es más, existe aplicaciones de todo tipo, forma, diseño y color que están disponibles para el usuario,

pero que, en los primeros teléfonos las aplicaciones más comunes eran las alarmas, calendarios, calculadoras, etc.

La gran mayoría de aplicaciones presenta los siguientes atributos (Roger S. Pressman, 2010) como se muestra en la Tabla 1:

| Atributos | Concepto |
|---|---|
| Uso intensivo de redes | Una aplicación reside en una red y debe atender las necesidades de una comunidad diversa de clientes. La red permite acceso y comunicación mundiales (por ejemplo, internet) o tiene acceso y comunicación limitados (por ejemplo, una intranet corporativa). |
| Concurrencia | Puede acceder un gran número de usuarios a la vez. |
| Carga impredecible | El número de usuarios de las aplicaciones cambian de un día a otro. Un día tal vez la utilicen cien personas, y otro quizá 10 000 usen el sistema. |
| Rendimiento | Si un usuario que hace uso de la aplicación debe esperar demasiado (para entrar, para el procesamiento), él o ella quizá decidan irse a otra parte. |
| Disponibilidad | Aunque no es razonable esperar una disponibilidad de 100%, es frecuente que los usuarios de las aplicaciones populares demanden acceso las 24 horas de los 365 días del año. |
| Orientadas a los datos | La función principal de muchas aplicaciones es el uso de hipermedias para presentar al usuario final contenido en forma de texto, gráficas, audio y video. |
| Contenido sensible | La calidad y naturaleza estética del contenido constituye un rasgo importante de la calidad de una aplicación. |
| Evolución continua | A diferencia del software de aplicación convencional que evoluciona a lo largo de una serie de etapas planeadas y separadas cronológicamente, las aplicaciones evolucionan en forma continua. |
| Inmediatez | Es una característica en muchos dominios de aplicación, es frecuente que las aplicaciones tengan plazos de algunos días o semanas para llegar al mercado. |
| Seguridad | Debido que las aplicaciones se encuentran disponibles con el acceso a una red, es difícil o imposible limitar la población de usuarios finales que pueden acceder a la aplicación. |

| | |
|---|---|
| Estética | Cuando se ha diseñado una aplicación para comercializar o vender productos o ideas, la estética tiene tanto que ver con el éxito como el diseño técnico. |

**Tabla 1:** Atributos

**Fuente:** Elaborado con contenido (Roger S. Pressman, 2010)

Actualmente en el mundo de los dispositivos móviles, ha existido un aumento de la demanda de desarrollo de aplicaciones móviles que posean ciertos atributos anteriormente mencionados, pues los usuarios están acostumbrados a usarlas en multitud tal es el ejemplo de Facebook, WhatsApp, Twitter, Instagram, etc. (Androides, 2017).

Todas estas apps se descargan e instalan desde las tiendas de aplicaciones, así también durante su uso es necesario actualizarlas, pero para ello, el usuario debe volver a descargarlas para obtener la última versión.

También tienen una característica muy interesante; y es el uso de las notificaciones del sistema operativo para mostrar al usuario avisos de la aplicación; de hecho, funciona como los mensajes que llegan del WhatsApp, cuando el teléfono no está siendo utilizado como se muestra en la Figura 3.


**Fig 3:** Notificaciones push

**Fuente:** propia

En definitiva, las aplicaciones nativas ofrecen una experiencia de uso más fluida que permite utilizar todas las características del teléfono, como la cámara, calendario, GPS, etc.; dependiendo del tipo de aplicaciones, ya sea de entretenimiento, sociales, utilitarias y productividad o educativas, favoreciendo la usabilidad y beneficio directamente al usuario.

### 1.1.3. Web Progresivas

Los usuarios últimamente buscan en la Web lo quizá podrían lograr a través de una app instalada en el teléfono, aunque dentro del proceso de buscar una aplicación correcta e instalarla para verificar la compatibilidad, el espacio que consumirá en el teléfono resulta ser muy tedioso (Fernández, 2016).

Las aplicaciones Web dieron un gran paso con la llegada de las aplicaciones Web Progresivas, que permiten realizar tareas directamente desde la Web posteriormente ya adaptadas a móviles, pero sin la necesidad de descargarlas e instalarlas, además, ofrece una experiencia de uso muy similar a la de una aplicación instalada en un teléfono móvil.

Las aplicaciones Web Progresivas, es un término que se da a una nueva generación de aplicaciones, cuya finalidad es incrementar la funcionalidad, de acuerdo a las capacidades del dispositivo en el que se ejecutan, incrementan, de ahí la palabra *progresiva*.

El nombre Web*,* hace referencia a que se construye utilizando estándares de desarrollo Web, capacidades avanzadas como las ofrecidas por HTML5 y los services workers(son archivos JS[6] que se ejecutan de forma separada con respecto a la página web y gestionan eventos lanzados bien por el navegador, bien por la propia web. ); y la parte final aplicación porque las aplicaciones Web Progresivas se comportan como Web nativas, pero usando tecnologías Web (Hernández, 2017).

Por ello, a medida que el usuario compila progresivamente una relación con la aplicación con el paso del tiempo, se hace más y más poderosa. Se carga rápidamente, incluso con redes débiles, envía notificaciones push relevantes, tiene un ícono en la pantalla principal y se carga como experiencia de pantalla completa y de primer nivel.

Asimismo, están disponibles para los usuarios a partir de la primera visita en una pestaña del navegador y no requieren instalación. A continuación, se muestra las principales características de una aplicación Web progresiva (Fernández, 2016):

- **Progresiva:**
  Funciona para todos los usuarios, sin importar la elección de navegador, porque está construida con mejora progresiva como principio central.

[6] JS: Java Script

- **Adaptable:**
  Se adapta a cualquier factor de formulario, sea escritorio, móvil, Tablet o lo que venga en el futuro.
- **Independiente de la conectividad:**
  Esta mejorada con service workers para trabajar sin conexión o con redes de mala calidad.
- **Estilo app:**
  Al usuario le parece una aplicación con interacciones y navegación estilo aplicación, porque está construida con modelo de shell.
- **Fresca:**
  Siempre actualizada gracias al proceso de actualización de service worker.
- **Segura:**
  Emitida vía HTTPS para evitar intromisiones y para garantizar que el contenido no se haya manipulado.
- **Descubrible:**
  Se puede identificar como "app" gracias al manifiesto W3C y al alcance de registro de service workers, lo que permite que los motores de búsqueda la encuentren.
- **Posibilidad de volver a interactuar:**
  Facilita la posibilidad de volver a interactuar a través de funciones como notificaciones push.
- **Instalable:**
  Permite a los usuarios "conservar" las apps que les resultan más útiles en su pantalla principal sin la molestia de una tienda de app**.**
- **Vinculable:**
  Puede compartir fácilmente vía URL, no requiere instalación compleja.

En consecuencia, existe varios beneficios que ofrecen las aplicaciones Web Progresivas entre ellos ser ejecutadas en cualquier sistema operativo mejorando el tiempo de carga y rendimiento de la página web (Rand, 2016).

## 1.2. Evolución histórica de las Arquitecturas Tecnológicas en la construcción de Web Progresivas

### 1.2.1. Etapa1: Arquitectura Cliente-Servidor

La arquitectura cliente-servidor nació por la necesidad que tenían las organizaciones de realizar sus actividades más ágiles y eficientemente, así su personal se hacía más productivo; por ello era vital establecer una infraestructura de procesamiento de la

información para mejorar la toma de decisiones y proporcionar un mejor servicio a los clientes (José Guillermo Valle, 2005).

IBM define al modelo Cliente/Servidor. “Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios o cualquier otro recurso de un grupo de trabajo, a través de la organización.”

Por lo tanto, la arquitectura cliente servidor es una relación entre procesos corriendo en máquinas separadas, donde el servidor es el proveedor de servicios y el cliente es un consumidor de esos servicios, que a su vez interactúa por un medio que les permite el pase de mensajes de pedido de servicio y respuesta como se muestra en la Figura 4.

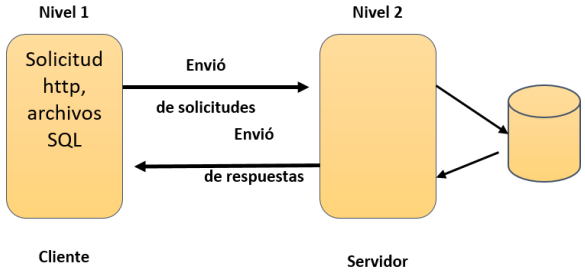



**Fig 4:** Arquitectura cliente-servidor

**Fuente:** propia, adaptada a https://bit.ly/2HbIbLz

Por tal motivo, la arquitectura cliente-servidor se comporta como un sistema multiusuarios que descentraliza la información y así pueda cumplir con cada tarea dentro del sistema, además de proveer una mayor escalabilidad, es decir, que se puede aumentar la capacidad de clientes y servidores por separado.

### 1.2.2. Etapa 2: Arquitectura en 3 Capas

La arquitectura en 2 capas dio lugar a la arquitectura en 3 capas debido a que el manejador de base de datos resultaba muy pequeño para migrarse o a su vez incorporar nuevos datos. En esta arquitectura, la lógica de la aplicación ocupa una capa intermedia que mejora el uso de sistemas que implementan un modelo de negocio como una tienda online (Gómez, 2017).

Por tanto, la arquitectura de tres niveles es un modelo que permite la distribución de la funcionalidad de la aplicación entre tres sistemas independientes.

La arquitectura en tres capas está compuesta en 3 partes diferenciadas como se muestra en la Figura 5, de tal manera que cada capa se comunique con la inferior. Esas 3 capas son las siguientes (Gómez, 2017):

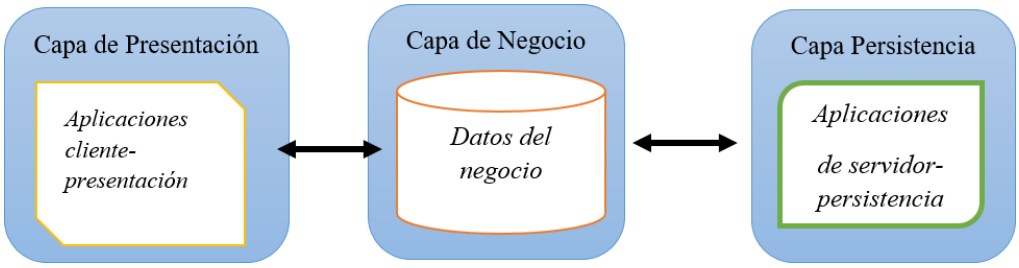



**Fig 5:** Distribución de las tres capas del modelo
**Fuente:** Propia

**a) Presentación:**

Es la que se encarga de que el sistema interactúe con el usuario y viceversa, muestra el sistema al usuario, le presenta la información y obtiene la información del usuario en un mínimo de proceso.

**b) Negocio:**

Es la encargada de gestionar la lógica de la aplicación, es decir; es donde residen las funciones que se ejecutan, se reciben las peticiones del usuario, se procesa la información y se envían las respuestas tras el proceso.

**c) Persistencia:**

Es la encargada de guardar los datos, y gestionar la base de datos. En consecuencia, la arquitectura de 3 capas permite una separación precisa de la interfaz de usuario de la lógica de la aplicación, esta división permite tener diferentes presentaciones accediendo a la lógica. Además, cada capa es un proceso independiente y definido, ejecutándose también en plataformas separadas. Por eso la arquitectura tradicional de tres capas, se instala una interfaz de usuario final en la computadora del Cliente.

### 1.2.3. Etapa 3: Arquitectura N Capas

Los desarrollos más recientes empiezan a experimentar con una capa adicional: Presentación; Aplicación; Dominio de la aplicación; Repositorio. La idea básica es separar todo lo que es programación de la aplicación de la presentación no hace cálculos, consultas o actualizaciones sobre el dominio. Es así que, en la arquitectura de 3 capas, cada servidor realiza una tarea especializada (un servicio). Por lo tanto, un servidor puede utilizar los servicios de otros servidores para proporcionar su propio servicio. Por ello, la arquitectura en 3 niveles es potencialmente una arquitectura en N niveles como se muestra en la Figura 6 (Alfsan, 2012).

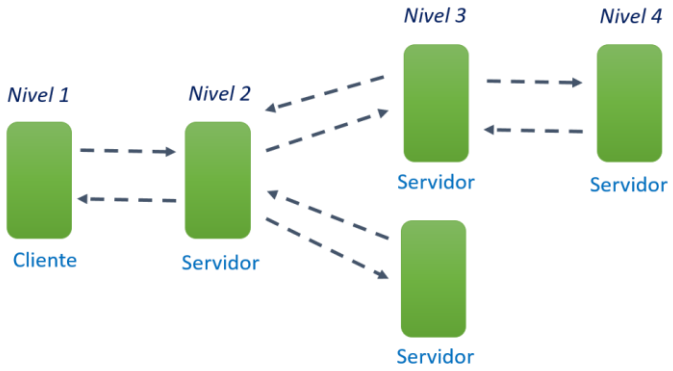



**Fig 6:** Arquitectura N capas
**Fuente:** propia, adaptada a https://bit.ly/2He3cWe

La arquitectura n capas es una arquitectura definida para la construcción de aplicaciones multiplataforma (Bermeo, 2012, pág. 25). Por lo tanto, proporcionan una gran cantidad de beneficios para aquellas empresas que necesitan solución flexible para resolver problemas de cambios constantes.

Está conformada por capas y niveles; las capas describe la parte lógica entre ellas los componentes y funcionalidades, sin embargo no saben la localización de ellas porque se pueden encontrar en diferentes servidores; en cuanto a los niveles están encargados de la distribución física de los componentes y funcionalidad en servidores separados, teniendo en cuenta la topología de redes y las localizaciones remotas (Ortiz, 2017).

Las capas están divididas en:

a) **Capa de presentación**

   Esta encargada de la parte visual de la aplicación, es decir es el formulario final que se presenta al usuario. (Santiago Domingo Moquillaza Henríquez, 2010).

b) **Capa de negocios**

   (Roger S. Pressman, 2010) refiere lo siguiente. Es la encargada del procesamiento desde la aplicación. Esta capa intermedia contendría los objetos que se corresponden con las entidades de la aplicación y la encargada de llevar la capacidad de mantenimiento y de reutilización.

c) **Capa de datos**

   Es la encargada de acceder a los datos para almacenar y recuperar la información del sistema (Santiago Domingo Moquillaza Henríquez, 2010).

A continuación, se muestra las principales características de la arquitectura:

- Los componentes de la aplicación pueden estar alojados en múltiples servidores permitiendo una mayor escalabilidad.
- Los problemas de limitación para las conexiones a las bases de datos se minimizan ya que la base de datos solo es vista desde la capa intermedia y no desde todos los clientes.
- Descomposición de los servicios de forma que la mayoría de interacciones ocurre solo entre las capas vecinas.

En consecuencia, la arquitectura N niveles facilita la presencia de sistemas distribuidos en los que se puede dividir los servicios y así aumentar la escalabilidad y mantenimiento.

### 1.2.4. Etapa 4: Arquitectura de Servicios (SOA)

Las empresas necesitan contar con sistemas integrados para poder interconectar todos los procesos, personas e información, tanto con la misma organización como con subsidiarias y socios comerciales.

Para lograr todo ese proceso, nace una herramienta basada en estándares para integrar sistemas y aplicaciones heterogéneos sobre una serie de plataformas y protocolos de comunicación con una metodología bien establecida, con el objetivo de lograr un nivel óptimo de integración, de manera que la infraestructura como se muestra en la Figura 7, facilite los cambios de acuerdo con las necesidades de la empresa. Este marco de trabajo conceptual es SOA (Arquitectura orientada a servicios) (Vera, 2013).

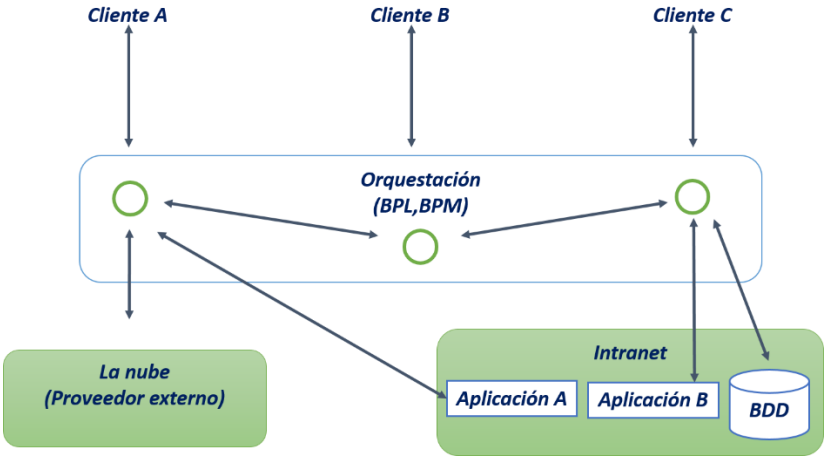



**Fig 7:** Arquitectura SOA
**Fuente:** propia, adaptada a https://bit.ly/2VsooLU

SOA es un marco de trabajo conceptual que establece una estructura de diseño para la integración de aplicaciones, que permite a las organizaciones unir los objetivos de negocio, en cuanto a flexibilidad de integración con sistemas legados y alineación directa a los procesos de negocio, con la infraestructura de TI[7].

Además, permite la reutilización de activos existentes para nuevos servicios que se puedan crear a partir de la infraestructura de TI. La arquitectura SOA tiene beneficios en la que distintos sistemas estén escrito en diferentes lenguajes de programación y que sean de distintas plataformas tecnológicas, puedan transformarse en servicios débilmente acoplados y altamente interoperables aumentando la eficiencia en los procesos (Cerpa, 2011, pág. 46).

De manera concisa utiliza una implementación denominada Servicios Web, pero servicio es un término muy utilizado en SOA; por ende, es necesario aclarar el significado de estos contextos.

**a) Servicio**

El termino servicio hace énfasis a que puede ser consumido por varios sistemas sin ser modificados, son exclusivamente para ser instalados y; de esta manera permanecer disponibles sin consumir recursos (Mejía, 2007, pág. 11). De tal manera un servicio es un componente reutilizable dentro de un proceso de negocio, con el propósito de proporcionar información.

**b) Servicios Web**

Los servicios Web son tecnologías que utilizan un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones (Vera, 2013).

Los servicios Web se basan en un conjunto de estándares de comunicación, como son XML para la representación de datos, SOAP (Simple Object Access Protocol) para el intercambio de datos, y el lenguaje WSDL (Web Services Description Language) para describir las funcionalidades de un servicio Web (Mejía, 2007).

Por consecuente los servicios son autónomos, es decir cada servicio se mantiene, desarrolla y despliega independientemente, también al ser independientes de otros servicios pueden ser reemplazados o actualizados sin afectar las aplicaciones en las que se esté utilizando siempre y cuando la interfaz sea compatible.

[7] TI: Tecnología de la Información

### 1.2.5. Etapa 5: Arquitectura de Cache (App Shell)

La arquitectura de Shell de una aplicación separa la infraestructura central de la aplicación y la IU de los datos. Toda la infraestructura y la IU se almacenan en caché de forma local mediante un service workers como se muestra en la Figura 8, a fin de que la Progressive Web App[8] solo deba recuperar los datos necesarios en las cargas posteriores, en lugar de volver a cargar todo (Partners, 2017).

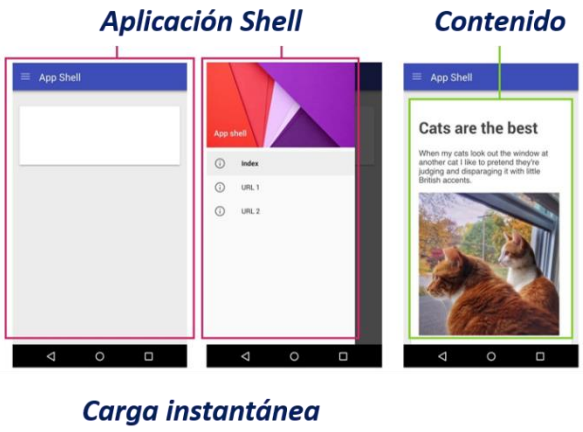



**Fig 8:** Arquitectura App Shell
**Fuente:** propia, adaptada a: https://bit.ly/2EwIEoy

El Shell de la aplicación es el HTML, CSS y JavaScript mínimos necesarios para impulsar la interfaz de usuario de una aplicación Web progresiva y es uno de los componentes que garantiza un rendimiento bueno y confiable.

La carga de la aplicación es rápida y capaz de almacenarse en caché inmediatamente. El almacenamiento en cache significa que los archivos de Shell se cargan una vez a través de la red y luego se guardan en el dispositivo local (Fernández, 2016). Es decir que cada vez que el usuario abre la aplicación, los archivos de Shell se cargan en la caché local del dispositivo, lo que resulta en tiempos de inicio muy rápidos.

## 1.3. Norma ISO

Son un conjunto de normas orientadas a ordenar la gestión de una empresa en sus distintos ámbitos. La alta competencia internacional acentuada por los procesos globalizadores de la economía y el mercado y el poder e importancia que ha ido

[8] App: Aplicaciones

tomando la Figura y la opinión de los consumidores, ha propiciado que dichas normas, ganen un reconocimiento y aceptación internacional (Peña, 2015).

Las normas ISO son establecidas por el Organismo Internacional de Estandarización (ISO), y se componen de estándares y guías relacionados con sistemas y herramientas específicas de gestión aplicables en cualquier tipo de organización.

## 1.4. Norma ISO 42010- Sistemas de ingeniería de software - Descripción Arquitectura

ISO / IEC / IEEE 42010 es una norma que define los requisitos para la designación de las arquitecturas de un sistema, del software y de la empresa cuyo objetivo es estandarizar la práctica de la descripción de la arquitectura, mediante la definición de términos estándar, presentando una base conceptual para la expresión, la comunicación y la revisión de la misma (Rodríguez, 2018).

Después de su predecesor, IEEE Std 1471, la norma hace una distinción estricta entre arquitecturas y descripciones de Arquitectura.

A continuación, es definido el modelo conceptual que establece la norma, la relación entre vista arquitectónica y punto de vista arquitectónico y en qué consiste un modelo arquitectónico.

### 1.4.1. Conceptos de la Norma ISO 42010

La primera parte de esta norma define un modelo conceptual que proporciona una sintaxis general y la semántica para descripciones de la arquitectura la cual se muestra en la Tabla 2.

| Término | Descripción |
|---|---|
| (Architecting)-Arquitectura | Proceso de concebir, definir, expresar, documentar, comunicar, certificar, mantener y mejorar una arquitectura a lo largo del ciclo de vida de un sistema (es decir, "Diseño"). |
| Arquitectura (Architecture) | Conceptos fundamentales o propiedades de un sistema en su ambiente, en sus elementos, en sus relaciones y en los principios de su diseño y evolución. |
| Descripción de Arquitectura («AD» Architecture Description) | Producto de trabajo usado para expresar una arquitectura. |
| Lenguaje de descripción de arquitectura (abreviatura «ADL) | Cualquier forma de expresión para su uso en las descripciones de la arquitectura. |

| | |
|---|---|
| Arquitectura Marco (Architecture Framework): | Convenciones, principios y prácticas para la descripción de arquitecturas, establecido dentro de un dominio específico de aplicación y / o la comunidad de partes interesadas (Stakeholders). |
| Punto de vista de la arquitectura (Architecture Viewpoint): | Producto de trabajo que establece los convenios para la construcción, interpretación y uso de vistas de arquitectura para enmarcar las preocupaciones específicas del sistema. |
| Vista de la arquitectura (Architecture View): | Producto de trabajo que expresa la arquitectura de un sistema desde el punto de vista de las preocupaciones (concern) específicas del sistema. |
| Preocupación (concern) | Interés en un sistema correspondiente a uno o más interesados en el sistema (stakeholders). |
| Clase de modelo (Model Kind) | Convenciones para un tipo de modelado. Una vista de la arquitectura se compone de varios modelos. |
| Interesados (Stakeholders) | Es un sistema individual, de equipo, de organización, el cual tiene un interés en un sistema. Los stakeholders pueden ser:<br>• **Clientes**: pagan por el sistema.<br>• **Usuarios**: utilizan el sistema.<br>• **Consultores**: prestan un bien o servicio a la entidad.<br>• **Patrocinadores**: financian todo o parte del proyecto.<br>• **Miembros del equipo de desarrollo:** líder de proyecto, analistas, programadores, testers. |
| Medio ambiente: (Enviroment) | Contexto del sistema para determinar la configuración y las circunstancias de todas las influencias sobre un sistema. |

**Tabla 2:** Terminología de la Norma ISO 42010

**Fuente:** Elaborado con contenido de la Norma ISO 42010 (IEEE, 2011)

La Figura 9 muestra los conceptos relacionados al aplicar esta norma internacional para producir una descripción de la arquitectura que se expresa para un sistema de interés. En esta Norma Internacional, el término Sistema de interés hace referencia al sistema cuya arquitectura está bajo consideración en la preparación de una descripción de la arquitectura.

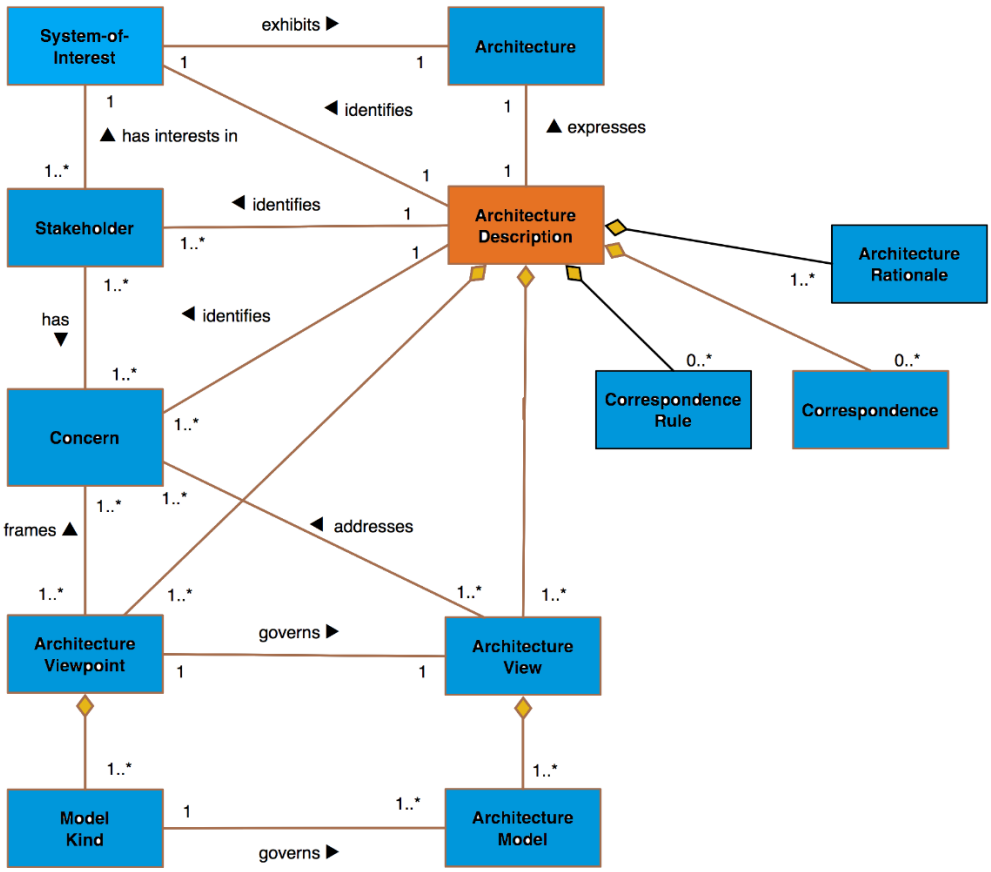



**Fig 9:** Modelo conceptual de la arquitectura

**Fuente:** (IEEE, 2011, pág. 5)

## 1.5. Caracterización gnoseológica de la gestión de pedidos

La gestión de pedidos consiste en actividades que resultan del cumplimiento de órdenes de pedido del cliente con la finalidad de mejorar el servicio al cliente. Sin embargo, para la apropiada gestión de pedidos y un correcto funcionamiento del restaurante, es necesario contar con un sistema de procedimientos operativos; los mismos que permitan el desarrollo de los diferentes procesos en forma ordenada de cada uno de las áreas con las que cuenta el establecimiento.

El manual de funciones o de procedimientos ayuda a tomar en cuenta los siguientes aspectos (Molina, 2014) como se muestra en el Anexo 1:

a) Seguridad Alimentaria
b) Higiene
c) Conocimientos de cocina
d) Actividades desarrolladas de cada uno de los trabajadores

Por tal motivo, con la información que brinda el manual de funciones que se encuentra anexado se realiza un flujograma de procesos de acuerdo a la gestión de pedidos como se muestra en la Figura 10.

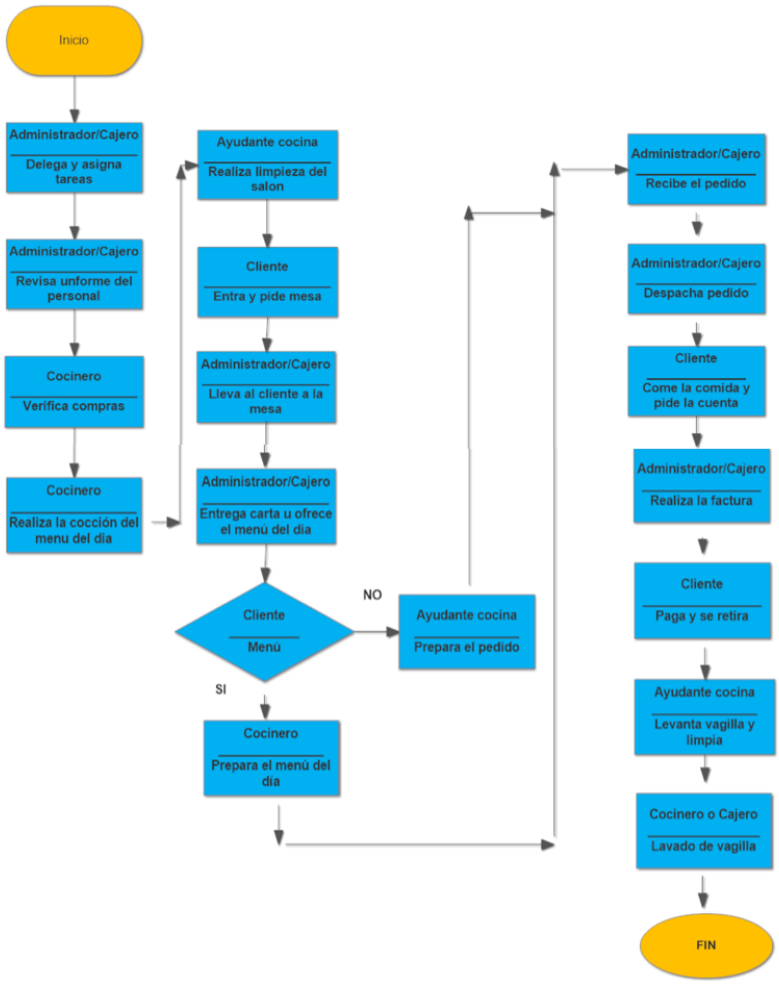



**Fig 10:** Flujograma de proceso de gestión de pedidos

**Fuente:** propia

## 1.6. Caracterización de una Arquitectura en la construcción de Web Progresivas

En la actualidad, el software está presente en gran cantidad de objetos que nos rodean: desde los teléfonos y otros dispositivos que llevamos con nosotros de forma casi permanente, hasta los sistemas que controlan las operaciones de organizaciones de toda índole o los que operan las sondas robóticas que exploran otros planetas.

Uno de los factores clave del éxito de los sistemas es su buen diseño; de manera particular, el diseño de lo que se conoce como arquitectura de software. (Maceda & Careaga, 2016)

Por ende, la construcción de aplicaciones Web Progresivas esta enfatizada en mejorar la experiencia del usuario final en el uso de la aplicación, es así que, necesita de una arquitectura lo bastante estructurada que permita homogenizar la interfaz y además

de proveer nuevas técnicas que apoyen en las nuevas APIs[9] para sacar mayor partido al desarrollo.

La arquitectura deseada debe permitir las siguientes características en una aplicación Web Progresiva como se muestra en la Figura 11:

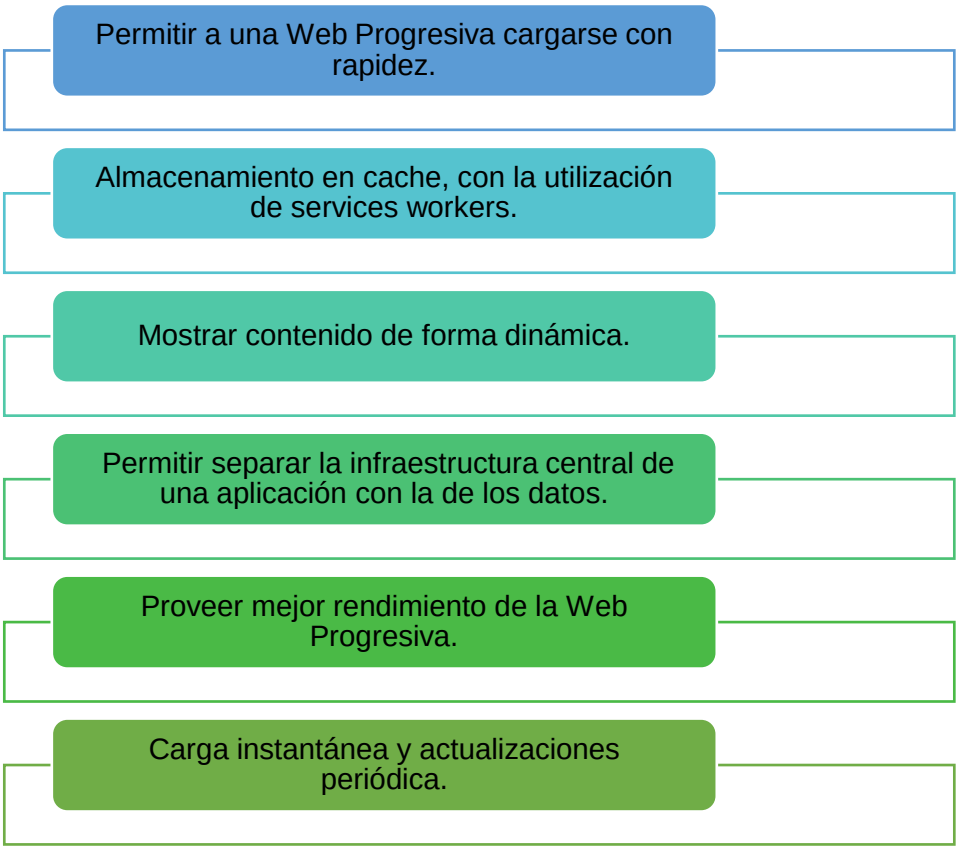



**Fig 11:** Características de la web progresiva

**Fuente:** propia

[9] APIs: Interfaz de programación de aplicaciones es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.

# CAPÍTULO 2

## Desarrollo de la investigación

El desarrollo está basado en el ciclo de vida que proporciona Cervantes por ello en la investigación se establecerá métricas medibles para determinar que arquitectura será la mejor opción para la construcción de Aplicaciones Web Progresivas, donde las arquitecturas a tratar son las siguientes App Shell, Responsive Design y SOA, mediante la norma ISO 42010 que hace referencia a la descripción de arquitecturas.

### 2.1. Ciclo de vida de la Arquitectura

Dentro del creación de sistemas existe un ciclo de vida para su realización utilizando el patrón de diseño MVVM (Modelo Vista VistaModelo), de manera similar se cuenta con un ciclo de desarrollo de una arquitectura de software que engloba actividades particulares (Maceda & Careaga, 2016). Por consiguiente, se conlleva a la elección favorable de la arquitectura para su posterior implementación. De esta manera

Esta se describe a continuación y a su vez se integran a las actividades técnicas del desarrollo de sistemas, como se muestra en la Figura 12.

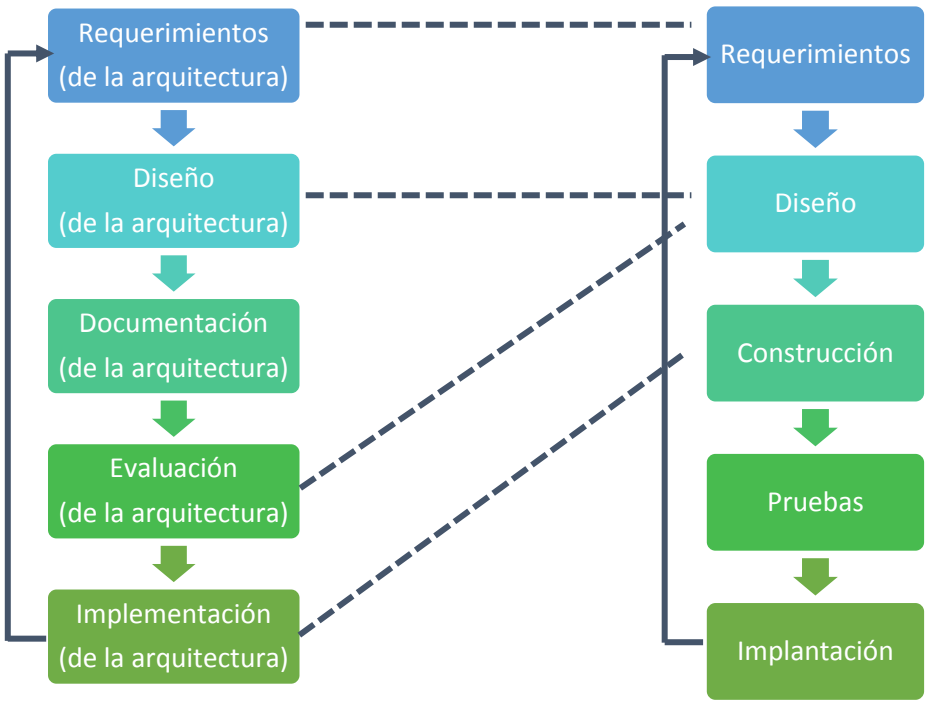



**Fig 12:** Actividades asociadas al ciclo de desarrollo de la arquitectura (a la izquierda) y su mapeo dentro de las actividades técnicas del desarrollo de sistemas (a la derecha).
**Fuente:** (Maceda & Careaga, 2016)

### 2.1.1. Requerimientos

En el ámbito de la Ingeniería de Software un requerimiento es una especificación que describe alguna funcionalidad, atributo o factor de calidad de un sistema de software (Maceda & Careaga, 2016).

Por ello, los requerimientos de software pueden describir el comportamiento, un atributo o propiedad del sistema, e incluso restringir la forma en que se construye el sistema.

Por otro lado, los requerimientos van de la mano con los objetivos de negocio del sistema. Por ejemplo, el objetivo de la microempresa “Deli Café Restaurant” dedicada a la venta de comida extranjera y nacional es *brindar un buen servicio a sus clientes*, que puede satisfacerse a través de una carta digital que pueda ser observada a través de navegadores web, Tablet o móvil.

Esta etapa se enfoca en la captura, documentación y priorización de requerimientos que influyen sobre la arquitectura y que, por lo habitual, se conocen en inglés como drivers arquitectónicos (Maceda & Careaga, 2016).

### 2.1.2. Tipos de Requerimientos

Antes de especificar los tipos de requerimientos que existen se debe tomar en cuenta que no todos los requerimientos para el desarrollo de un sistema son los mismos, es decir que, el dueño, el cliente y el desarrollador tienen diferentes intereses. Por ejemplo, realizar cambios en un sistema, es un atributo de calidad más relevante para los desarrolladores que para los usuarios finales.

Por ejemplo, para poder incrementar los ingresos es un requerimiento para el dueño pero que es de menor importancia para los clientes y desarrolladores. Por lo tanto, para que se pueda facilitar la comprensión de cada requerimiento es necesario clasificarlos por tipos y niveles propuesto por (Wiegers & Beatty, 2013) como se muestra en la Figura 13.

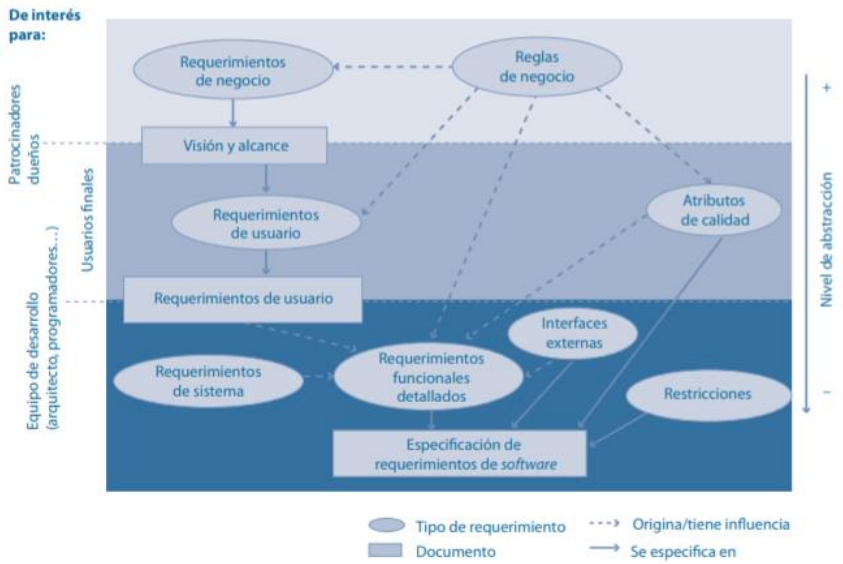



**Fig 13:** Tipos de requerimientos
**Fuente:** (Wiegers & Beatty, 2013)

De acuerdo a (Wiegers & Beatty, 2013) el nivel superior corresponde a tipos de requerimientos que describen aspectos del negocio. El nivel medio habla de cuestiones referidas a la interacción con los usuarios. Por último, el nivel inferior corresponde a los requerimientos que describen situaciones o elementos detallados que se necesitan para realizar el diseño del sistema tal como se observa en la Tabla 3.

| Término | Descripción |
|---|---|
| Requerimientos de negocio | Objetivos de alto nivel del negocio, que la organización espera lograr con la implementación del sistema. |
| Reglas de negocio | Política, guía, estándar que define o restringe algún aspecto del negocio. |
| Requerimientos de usuario | Describen metas o tareas que el usuario debe ser capaz de realizar con el sistema, incluyen la descripción de atributos o características con las que va a contar el sistema para la satisfacción del usuario. |

| | |
|---|---|
| Requerimientos de sistemas | Es un tipo de requerimiento que contiene múltiples subsistemas, que puedan estar formados por software o hardware |
| Requerimientos funcionales | Es la descripción que un sistema debe tener bajo condiciones específicas. Describen lo que los desarrolladores deben de realizar para cumplir con los requerimientos de usuario. |
| Interfaces externas | Descripción de una conexión entre un sistema de software y un usuario, otro sistemas, o un dispositivo de hardware |
| Atributos de calidad | Un tipo de requerimiento no funcional que describe una característica o propiedad de un sistema, tales como la disponibilidad, usabilidad, seguridad, rendimiento, portabilidad. |
| Restricciones | Imposiciones sobre el diseño y construcción de un sistema, que deben ser respetadas por el desarrollador. |

**Tabla 3:** Tipos de información de requerimientos de software

**Fuente:** (Rea, 2017)

Por lo tanto, al tener esta especificación de requerimientos no habrá dificultad en recolectarlas, además de que así ayuda a proveer de los documentos dónde generalmente se especifican estos requerimientos

### 2.1.3. Documentos Visión y Alcance- Contexto del negocio

#### a. Estructura organizativa

La misión y visión de la empresa son los siguientes:

#### b. Objetivos de negocio

| ID | Descripción del objetivo de negocio |
|---|---|
| ON-1 | Incrementar en un 10% el número de comensales mediante la carta digital. |
| ON-2 | Incrementar en un 20% los ingresos mensuales. |

| | |
|---|---|
| ON-3 | Proporcionar mecanismos que permitan integrar la aplicación a terceros que requieran la aplicación web progresiva. |
| ON-4 | Liberar el sistema antes del 30 de Mayo del 2019. |

**Tabla 4:** Objetivos de negocio

**Fuente:** Propia

### 2.1.4. Visión de la solución

#### a. Fase de visión

La presente aplicación de gestión de pedidos permitirá por un lado la visualización de la información acerca del restaurante "Deli Café Restaurant" y por otro lado tendrá una carta digital donde se encontrará toda la variedad del menú del restaurante y que será accesible por medio de un navegador web a través de cualquier dispositivo.

#### b. Características de la aplicación

| ID | Descripción | Prioridad | Objetivos del negocio asociado |
|---|---|---|---|
| CAR-01 | La aplicación debe permitir ver la información general acerca del Restaurante. | Media | ON-1 |
| CAR-02 | La aplicación debe permitir trabajar de forma online y offline. | Alta | ON-1, ON-2 |
| CAR-03 | La aplicación debe integrarse con redes sociales (Facebook, Instagram) | Baja | ON-1 |
| CAR-04 | La aplicación debe permitir recibir notificaciones | Alta | ON-2 |

| | | | |
|---|---|---|---|
| CAR-06 | La aplicación debe estar disponible 24 horas al día . | Media | ON-3 |
| CAR-07 | La aplicación debe garantizar ser instalable sin consumir recursos. | Alta | ON-2 |
| CAR-08 | La aplicación debe proporcionar una interfaz sencilla y fácil de manejar por los usuarios. | Alta | ON-1 |

**Tabla 5:** Características de la aplicación

**Fuente:** Propia

### 2.1.5. Alcance

| Número de entrega | Tema principal | ID de características |
|---|---|---|
| 1.0 | Funcionalidad básica | CAR-01, CAR-02, CAR-04, CAR-05,CAR-07, CAR-08,CAR-09 |
| 2.0 | Estabilidad del sistema | CAR-06 |
| 3.0 | Visibilidad | CAR-03 |

**Tabla 6:** Alcance

**Fuente:** Propia

### 2.1.6. Drivers Arquitectónicos

Dentro de los requerimientos que se consideran para el desarrollo de un sistema y que se derivan de los objetivos de negocio, existe un subconjunto que tiene una gran importancia relativa a la arquitectura que se les conoce como *drivers arquitectónicos* o Conductores de la Arquitectura.

Por consecuente, la etapa de requerimientos está centrada en la identificación, documentación y priorización de drivers (Maceda & Careaga, 2016). Está clasificada en tres clases como se muestra en la Tabla 7:

| Término | Descripción |
|---|---|
| Drivers funcionales | Subconjunto de los requerimientos funcionales que proveen información relevante para llevar a cabo la descomposición funcional del sistema y asignar estas funcionalidades a elementos específicos en la arquitectura.<br><br>Además, se considera que los drivers funcionales deben satisfacer los objetivos del negocio del sistema. |
| Drivers de atributos de calidad | Por lo habitual, todos los requerimientos de atributos de calidad son drivers de la arquitectura, independientemente de su prioridad. Sin embargo como el tiempo para realizar el diseño arquitectónico es acotado, para producir un diseño inicial es preferible considerar nada más un subconjunto de los requerimientos de atributos de calidad. |
| Drivers de restricciones | Expresan aspectos que deben considerarse al realizar el diseño y limitan las decisiones que se pueden tomar, aunque los requerimientos de este tipo no tienen prioridad. |

**Tabla 7:** Drivers arquitectónicos

**Fuente:** Elaborado con contenido de "Arquitectura de Software" por (Maceda & Careaga, 2016)

Todos los requerimientos mencionados en la Tabla 7 están bien formalizados y priorizados por los stakeholders o parte interesada.

### 2.1.7. Recolección de drivers arquitectónicos

Para la extracción de los drivers arquitectónicos como primer paso es necesario tener un documento de visión y alcance por el cual desarrollar la aplicación Web, de manera que resulte utilizarlo como base para la identificación de los drivers.

### 2.1.8. Drivers funcionales

Para la recolección de los drivers funcionales se lo realiza usando la técnica de casos de uso.

#### a. Modelo de caso de usos

A continuación, se muestra el modelo de casos de uso de la aplicación, usando un diagrama de uml como se muestra en la Figura 14, con una descripción de los mismos como se muestra en la Tabla 8.

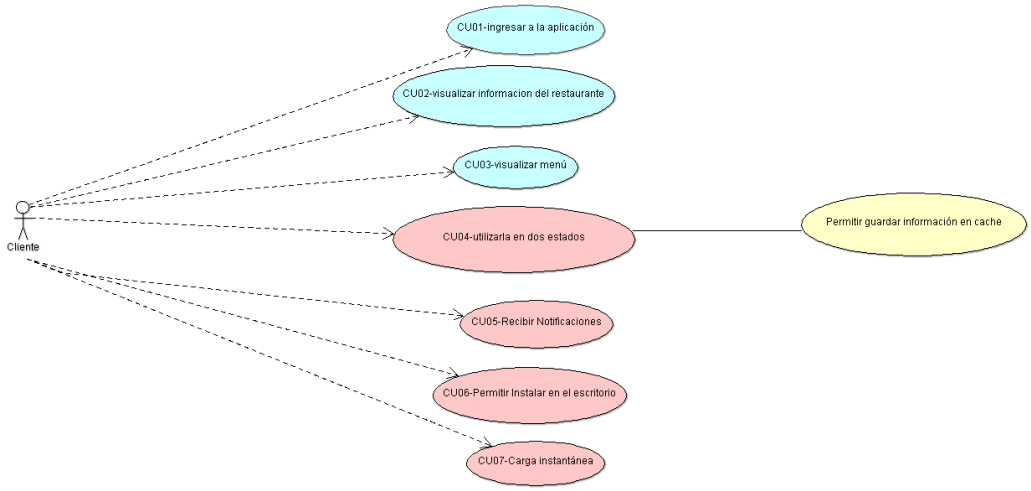



**Fig 14:** Casos de uso de la aplicación

**Fuente:** Propia

| ID | Descripción | Característica asociada |
|---|---|---|
| CU02 | Permite al usuario observar la información que existe del restaurante. | CAR-01, CAR-06 |
| CU03 | Permite al usuario observar la información de todo el menú | CAR-05 |
| CU04 | Permite al usuario utilizar la aplicación de dos estados , es decir de forma online y offline | CAR-02 |
| CU05 | Permite al usuario recibir notificaciones de la app | CAR-04 |
| CU06 | Permite al usuario instalar la app si consumir memoria en el móvil o pc. | CAR-07 |

**Tabla 8:** Casos de uso

**Fuente:** Propia

### 2.1.9. Drivers de atributos de calidad

A continuación, se muestra desglosado los atributos de calidad. La priorización se realiza en este caso considerando dos dimensiones que pueden tomar valores alto, medio y bajo, como se muestra en la Tabla 9.

| ID | Categoría | Escenario | Prioridad |
|---|---|---|---|
| EAC-01 | Desempeño | **Fuente de estímulo:** un cliente<br><br>**Estímulo:** El usuario ingresa a la aplicación y pierde conexión a la red.<br><br>**Artefacto:** la aplicación (CU-04).<br><br>**Entorno:** en un momento normal de operación con 25 clientes y se pierde la conexión a internet.<br><br>**Respuesta:** La aplicación guarda las peticiones en segundo plano de los servidores sin que el usuario sea consciente.<br><br>**Medida de respuesta:** un tiempo no mayor a tres segundos. | (A, A) |
| EAC-02 | Disponibilidad del sistema | Al ocurrir una falla interna del sistema que detiene su operación en el momento que estaba funcionando de manera normal. La aplicación vuelve a operar normalmente en un tiempo no mayor a 5 minutos. | (M,A) |
| EAC-03 | Seguridad | Un atacante al interceptar la comunicación entre el cliente y la aplicación mientras realiza su pedido en un momento normal de la operación. El atacante no puede obtener ninguna información. | (A, B) |
| EAC-04 | Facilidad de prueba | Al realizar pruebas, la arquitectura debe permitir separar la infraestructura central de la app y la IU de los datos. | (B,B) |
| EAC-05 | Modificabilidad | Podemos dejar un hilo secundario comprobando cuando exista cambios en la | (B,B) |

| | | | |
|---|---|---|---|
| | | aplicación y que esta los instale cuando los haya. | |

**Tabla 9:** Drivers de atributos de calidad

**Fuente:** Propia

### 2.1.10. Drivers de restricciones

A continuación, se muestra la lista de restricciones como se muestra en la Tabla 10 asociadas a la aplicación. Se identifican dos categorías de estas: a) las que expresa el cliente de forma explícita, que en este caso se derivan del documento de visión y alcance, y b) las que son parte del desarrollo, y que pueden no aparecer en este documento. (Maceda & Careaga, 2016).

| Tipo de restricción | Descripción |
|---|---|
| Del cliente | Tiempos de entrega. |
| Del desarrollador | El desarrollador está familiarizado con el framework Angular6.<br><br>Uso de librerías y herramientas Open Source. |

**Tabla 10:** Restricciones

**Fuente:** Propia

## 2.2. Diseño de la arquitectura

Según (Ralph & Wand, 2009) el diseño se atribuye a una especificación de un objeto, manifestada por un agente con el fin de lograr los objetivos planteados, en un entorno en particular, utilizando un conjunto de componentes que satisfacen un conjunto de requisitos sujetos a restricciones. Por tal motivo, para la creación del diseño se realizará de la toma de decisiones que se realice con respecto a las necesidades del usuario.

Dentro del diseño de la arquitectura y con respecto al concepto antes mencionado se detalla lo siguiente:

a. El objeto está relacionado con las distintas estructuras que componen la arquitectura.

b. El Agente se refiere al arquitecto u otros encargados del diseño.

c. Los objetivos son los requerimientos que influyen a la arquitectura es decir los drivers.

d. El entorno se refiere al uso del sistema por parte de los usuarios finales, como al ambiente en que se desarrolla la aplicación.

e. El conjunto de requisitos incluye la lista de drivers como los funcionales y no funcionales (atributos de calidad).

f. Las restricciones son todas las limitaciones impuestas por el cliente o por parte de los desarrolladores.

### 2.2.1. Métodos de diseño de la arquitectura

La norma ISO 42010 indica que para el diseño de una arquitectura empresarial es necesario la intervención de diversos métodos como los que se muestran a continuación.

a) Método diseñado por atributos (ADD. siglas en ingles).

b) Métodos de diseño centrado en la arquitectura (ACDM).

c) Método de definición de arquitectura de Viewpoints y Perspectives.

Para el caso de estudio se opta por el método diseñado por atributos o **ADD** como se muestra en la Figura 15, que tiene como enfoque la entrada de una lista de drivers estructurales *con* el objetivo de crear a su salida un conjunto de estructuras que conformaran al diseño de la arquitectura. A continuación, se muestra los pasos para el diseño correcto de la arquitectura.

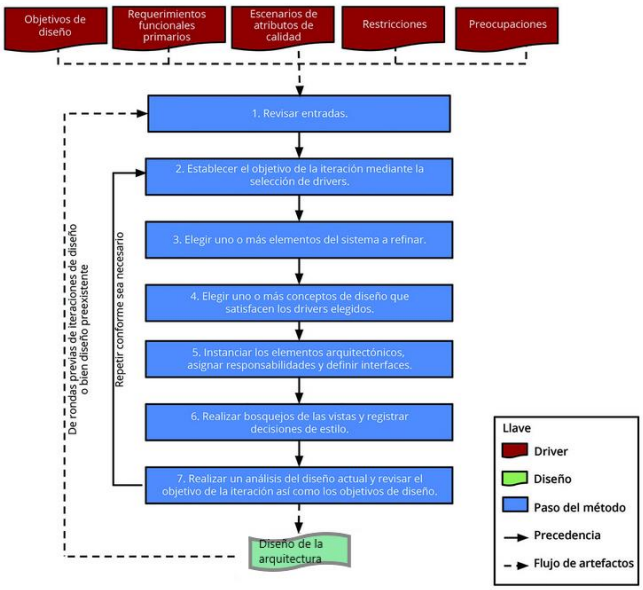



**Fig 15:** Pasos del método de diseño ADD

**Fuente:** propia, adaptada a: https://bit.ly/2EKCR0b

### 2.2.2. Aplicación del Método ADD

a) **Paso 1:** Revisar que se tiene suficiente información sobre los drivers arquitectónicos.

En este paso lo primordial es asegurarse que se tenga los datos suficientes sobre los drivers arquitecturales. Los drivers están realizados en la etapa de Requerimientos *de la Arquitectura,* siendo así la primera iteración.

b) **Paso 2:** Elegir un elemento del sistema a descomponer.

El elemento puede ser el sistema completo, si es un desarrollo nuevo, o un elemento obtenido de una iteración anterior. En este caso es la Aplicación Web Progresiva para fortalecer la gestión de pedidos.

c) **Paso 3:** Identificar drivers arquitectónicos asociados al elemento.

En esta etapa se elige una parte del conjunto inicial de drivers y se relacionan con el elemento elegido. Para este caso se dispone del driver principal detallado en el **CU04** como se muestra en la Tabla 11:

| | |
|---|---|
| **Elementos a descomponer** | Las distintas capas de la aplicación |
| **Drivers elegidos para la iteración** | Caso de uso primario:<br>CU4: Permite al usuario utilizar la aplicación de dos estados , es decir de forma online y offline. |

**Tabla 11:** Elemento a descomponer
**Fuente:** Propia

**Caso de uso primario:** Permite al usuario utilizar la aplicación de dos estados, es decir de forma online y offline

**Atributos de calidad:** Todos los atributos de calidad para la aplicación se detallan detenidamente en la Tabla 7 (Atributos de la Calidad). A continuación, se detalla uno de los atributos como es la Usabilidad- “Un usuario al estar interactuando con la aplicación, pierde conexión con la red en un momento normal de la operación. La aplicación guarda las peticiones en segundo plano de los servidores sin que el usuario sea consciente. Este cacheo puede ayudar a la fluidez de la aplicación hasta volver a tener una conexión a la red”.

**Restricción:** Uso de librerías y herramientas Open Source.

d) **Paso 4:** Elegir un concepto de diseño que satisfaga los drivers.

En este paso se selecciona un concepto general de diseño (ya sea patrones o tácticas) en relación con el elemento y los drivers elegidos. Este elemento es descompuesto mediante la aplicación de patrones asociados a tal concepto (Cervantes & Valencia, 2010).

Para este caso, el escenario principal es la usabilidad siendo así la iteración primordial del diseño para establecer una estructura de alto nivel. Por ello es necesario un patrón por capas para dividir a la app. Además, de optar por un diseño de módulos para almacenar los componentes y servicios de la aplicación.

Este patrón es el de MVMM significa Modelo Vista VistaModelo se aplica al framework Angular Versión 6 que es open Source. El MVMM es un patrón de diseño que separa los datos de la aplicación de la interfaz de usuario esto hace que la forma de sincronizar los datos entre la vista y el modelo-vista es totalmente dependiente, es decir, en la vista podemos modificar el modelo y en el modelo podemos modificar la vista. Siendo así, este modelo el que a través del driver seleccionado satisface los atributos usabilidad, desempeño, disponibilidad, seguridad y Modificabilidad.

e) **Paso 5**: Instanciar los elementos y asignar responsabilidades.

En esta fase se crean instancias de elementos derivados de los patrones y se describen sus responsabilidades (Maceda & Careaga, 2016).

Para el caso, se muestra en la Figura 16 la posible arquitectura que se adapte a la construcción de una Web progresiva, y de esta manera, verificar entre las arquitecturas: App Shell, SOA y Responsive Design cuál es la más idónea.

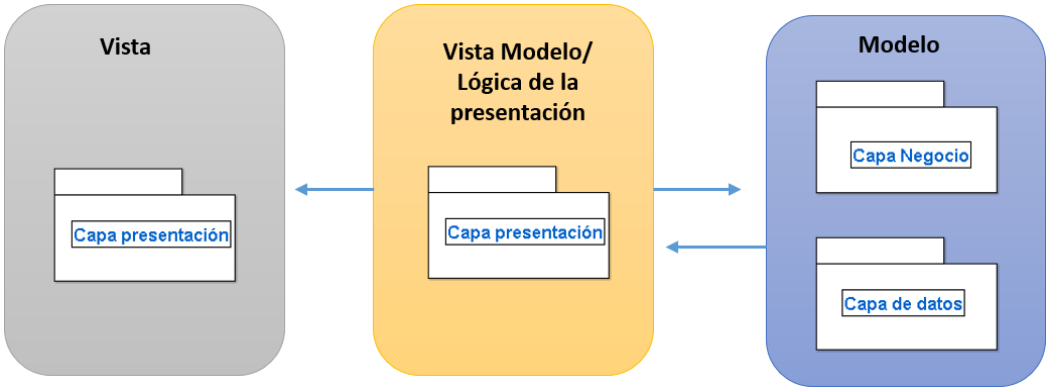



**Fig 16:** Perspectiva lógica UML

**Fuente:** Propia

Para la asignación de responsabilidades se detallarían de la siguiente manera:

- La capa presentación contendrá los componentes que permiten desplegar las pantallas asociadas de las interacciones del usuario.
- La capa de negocio contendrá los componentes encargados de llevar a cabo la ejecución de los casos de uso de la aplicación.
- La capa de datos contendrá los componentes encargados de almacenar todas las peticiones en un archivo.js y por ende permitir trabajar de forma online o offline.

**f)** **Paso 6**: Definir interfaces para los elementos instanciados.

En este paso se identifican propiedades para los elementos instanciados y, más específicamente, las interfaces de los mismos (Maceda & Careaga, 2016). A continuación, se define la interfaz de acuerdo al driver seleccionado como se muestra en la Figura 17.

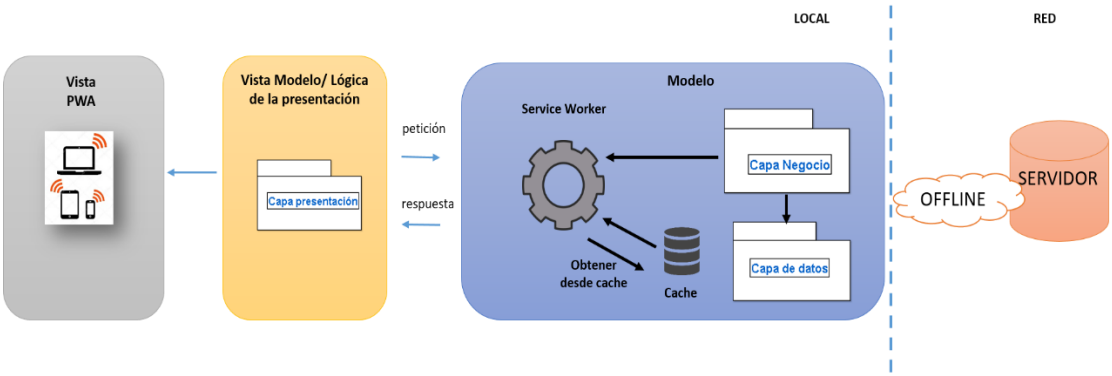



**Fig 17:** Perspectiva física UML
**Fuente:** Propia

| Elemento | Responsabilidad | Propiedades |
|---|---|---|
| PC del usuario (Vista) | Es un computador personal mediante el cual los usuarios acceden a la aplicación . | Navegador Firefox, Google Chrome |
| Dispositivo móvil (Vista) | Dispositivo por el cual accede el usuario. | Android y en futuro IOS. |
| Lógica de la presentación | Contiene la capa de presentación. | |

| | | |
|---|---|---|
| Modelo | Contiene la capa de negocio, en conjunto con un service worker y manifest.<br><br>Contiene los datos de la aplicación. | Servidor propio del framework. |

**Tabla 12:** Descripción del patrón

**Fuente:** Propia

g) **Paso 7:** Verificar y refinar requerimientos y transformarlos en restricciones para los elementos instanciados.

En este paso se verifica si se han satisfecho los drivers y, en caso necesario, se refinan y se asocian a los elementos identificados durante la iteración (Maceda & Careaga, 2016).

Por último, de acuerdo al driver seleccionado y con la información adquirida de la utilización del modelo ADD se muestra a continuación una perspectiva lógica UML del funcionamiento de la arquitectura como se muestra en la Figura 18.

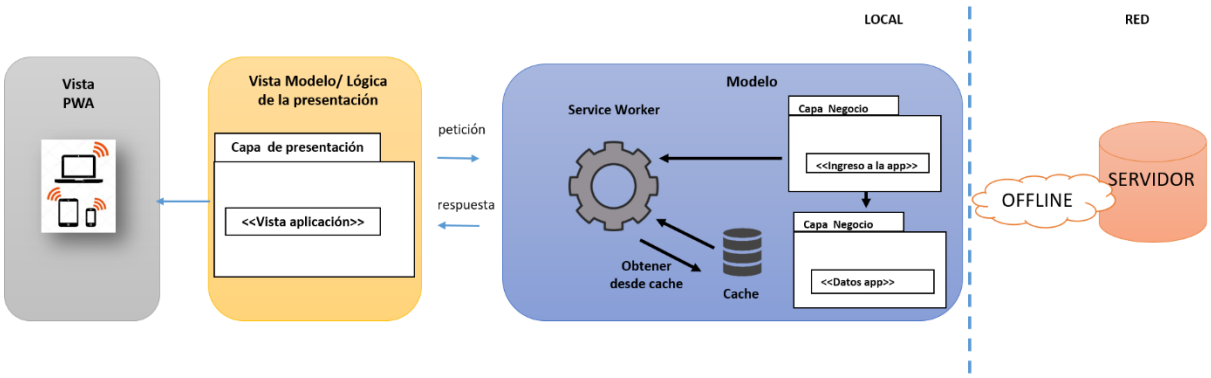



**Fig 18:** Perspectiva lógica UML del driver CU4

**Fuente:** Propia

| Elemento | Responsabilidad | Propiedades |
|---|---|---|
| Vista aplicación | Este componente es responsable de mostrar toda la información de la app **CU04**. | Lenguaje=Java<br><br>Framework= Angular V6 |
| Ingreso a la app | Este componente es responsable de proporcionar | Lenguaje=Java<br><br>Framework= Angular V6 |

| | | |
|---|---|---|
| | funciones para soportar el **CU04.** | |
| Datos app | En este componente se encuentra la información de la app en segundo plano (Cache). | |

**Tabla 13:** Descripción del funcionamiento de la aplicación

**Fuente:** Propia

El presente diagrama de secuencia UML muestra la manera en la que interactúan los componentes para soportar el driver CU-04.

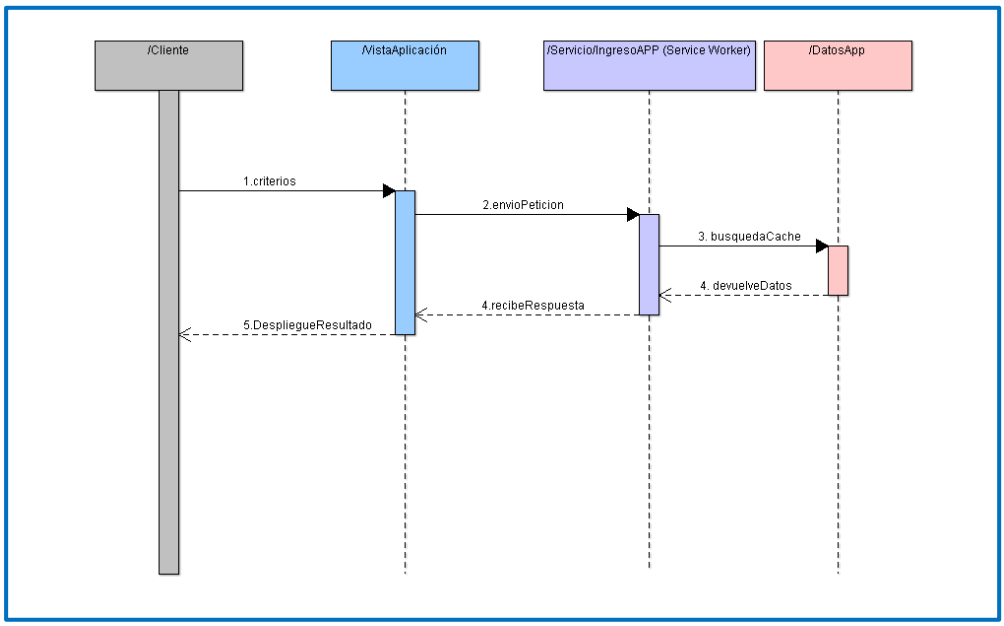



**Fig 19:** Diagrama de secuencia UML del driver CU4

**Fuente:** Propia

## 2.3. Documentación

### 2.3.1. Generar una lista de vistas candidatas

En la creación de la lista es necesario realizar una Tabla que contenga el conjunto de interesados en el sistema (IEEE, 2011), en las columnas las vistas que se detallan a continuación:

a. Vistas de módulos: describe las estructuras de la arquitectura.

- Descomposición: permite describir la estructura interna de los módulos.
- Uso: permite describir las dependencias funcionales entre módulos.

- Capas: permite detallar relaciones de grupos de los módulos llamados capas.
- Modelo de datos: permite detallar datos acerca de entidades que operen sobre el sistema.

b. Vistas de componentes y conectores: describen la arquitectura en términos de elementos visibles. Para el caso de la investigación de ha elegido App Shell debido a que separa la infraestructura central de la app y la IU de los datos.

c. Vistas de localización: permite detallar las relaciones entre los elementos documentados en otras vistas y los elementos físicos.

- Puesta en marcha-Implementación: indica los elementos del hardware como el pc de usuario, dispositivo móvil, lógica de la presentación y modelo.
- Instalación: describe la ubicación de las unidades de los módulos.
- Asignación de trabajo: describe la relación entre módulos sistemáticos y el personal que lo desarrolla.

La Tabla 14 muestra la información de nuestro caso de investigación. El conjunto de interesados se detalla del documento de visión y alcance anteriormente realizado, y que se lo realiza de acuerdo a las siguientes prioridades:

**D** = Muy detallada, **M** = Considerable detalle, **P** = Poco detalle.

La Tabla 14 se la realizo de acuerdo a las opiniones de los interesados

| | | Vistas de Módulos | | | | Vista de componentes y conectores | Vistas de localización | | | Otra documentación | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ROLES | Descomposición | Uso | Capas(MVVM) | Modelo de datos | App Shell Arquitectura | Puesta en marcha | Instalación | Asignación de trabajo | Documentación de interfaces | Diagramas de contextos | Descripción de decisiones arquitectónicas |
| Líder de proyecto | AG | m | m | m | m | | d | | d | | d | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Equipo de desarrollo | CB | d | d | d | d | d | m | d | | d | d | d |
| Equipo de pruebas | AQ | d | d | p | | d | m | m | | d | d | |
| Representante infraestructura | JG | m | m | d | d | m | d | d | | | | |
| Arquitecto | XH | d | d | | | d | d | d | d | | d | d |
| Representante usuarios | YE | | | | | p | p | | | | | |

**Tabla 14:** Lista de vistas candidatas

**Fuente:** Propia

De acuerdo a la Tabla 14 a la mayoría de las vistas está asignada un nivel de detalle alto porque la mayoría de los involucrados con la aplicación es el personal técnico que se detalla a continuación, consolidad a través de una Acta.

**AG** ALEXANDER GUEVARA, **CB** CARINA BAUTISTA, **AQ** ANTONIO QUIÑA, **ING** JOSÉ GUERRA, **XH** XAVIER HARO, **YE** YOLANDA ESPARZA

### 2.3.2. Combinar vistas

Una vez generada la Tabla de interesados como lo indica en la Norma ISO 42010 y las vistas y de acuerdo a las prioridades que se ha dado a cada se decide generar las vistas listadas a continuación:

Vistas de módulos:

- Vista1: Capas, Descomposición y Uso-muy detallado.
- Vista2: Modelo de datos

Vistas de comportamiento y localización

- App-Shell-muy detallado
- Implantación e instalación –muy detallada

### 2.3.3. Priorizar las vistas

Este es el último en la que el arquitecto da prioridades a las vistas, para este caso de investigación es necesario considerar la utilidad que tiene cada vista durante el desarrollo de la aplicación como se muestra en la Tabla 15.

| Prioridad | Nombre de vistas |
|---|---|
| 1 | Vista1: Capas(MVVM), Descomposición y Uso |
| | Vista2: Modelo de datos |
| 2 | Vista3: App-Shell |
| | Vista4: Implantación e instalación |

**Tabla 15:** Priorización de vistas

**Fuente:** Propia

Al estar priorizadas las vistas se debe documentar cada una de las vistas, a continuación, se muestra la documentación generada para la vista 3. App Shell, el CU-04 y los diferentes documentos que se empleó.

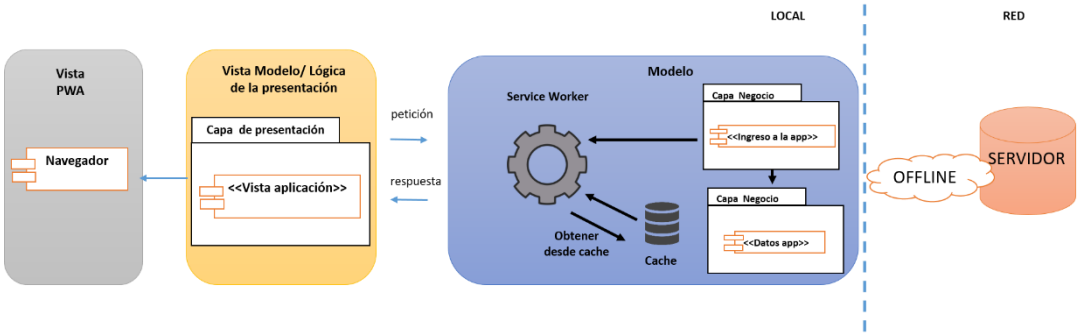



**Fig 20:** Documentación de la vista App Shell

**Fuente:** Propia

| | Decisión | Justificación |
|---|---|---|
| EAC-01 | Uso del patrón MVVM | Este patrón permite separar el código de la interfaz del usuario con la finalidad de hacer pruebas automatizadas sobre el resto del código y permitir mantener la carga inicial para mostrar solo el diseño de la página apenas se abra la app. |

**Tabla 16:** Descripción de decisiones arquitectónicas

**Fuente:** Propia

## 2.4. Evaluación

### 2.4.1. Análisis comparativo

A continuación, se detalla los indicadores que permitan satisfacer el desarrollo de una Web progresiva mediante iteraciones concurrentes de cada arquitectura estudiada, para ello se generó un laboratorio de pruebas controladas ver Anexo 2, aplicadas al Performance, Adaptabilidad y Funcionalidad, como se detalla en cada indicador:

a. **Funcionalidad:** Significa que los usuarios puedan acceder a la aplicación de forma online y offline. Para demostrarlo se realiza con la herramienta que ofrece Google Chrome al hacer clic derecho en la web, escoger la opción ***Inspeccionar,*** luego dirigirse a la opción Aplicación, activar la opción ***Offline*** como se muestra en la Figura 21. y recargar la página.

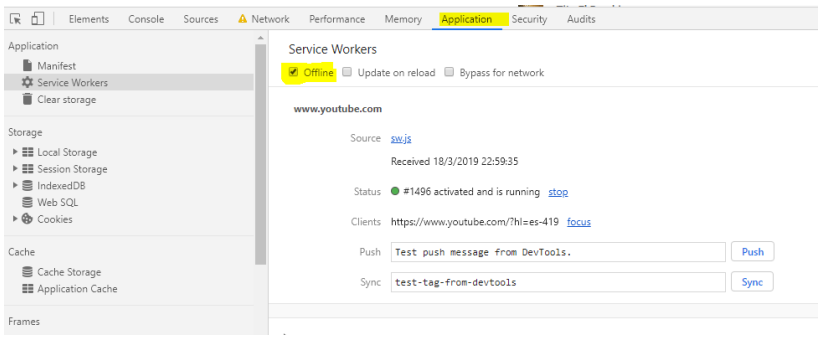



**Fig 21:** Trabajar sin red

**Fuente:** Propia

- **Responsive Design:** Según la prueba cuando se desconecta de la red a la aplicación deja de funcionar como se muestra en la Figura 22.

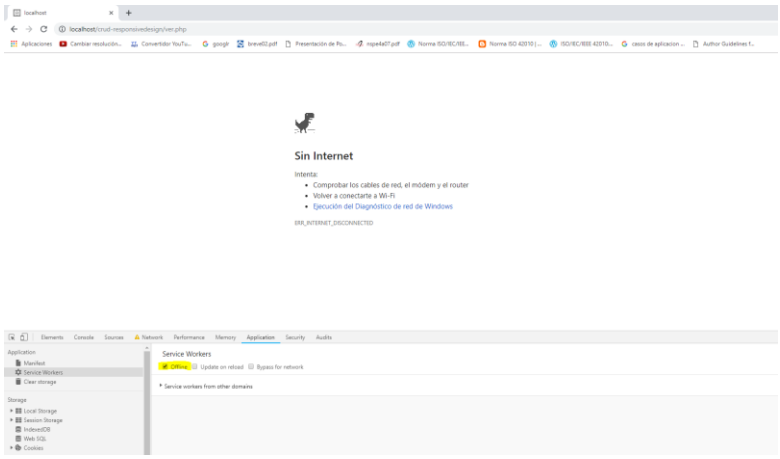



**Fig 22:** Arquitectura Responsive Design-Funcionalidad

**Fuente:** Propia

- **SOA:** Según la prueba cuando se desconecta de la red a la aplicación no funciona se muestra en la Figura 23.

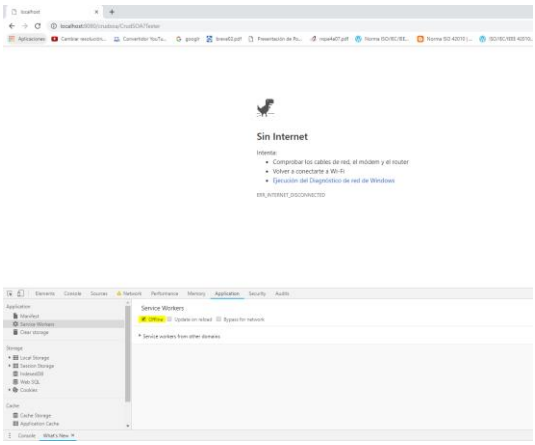



**Fig 23:** Arquitectura SOA-Funcionalidad

**Fuente:** Propia

- **App Shell:** Según la prueba permite que la aplicación funcione sin conexión a internet puesto que trabaja en segundo plano como se muestra en la Figura 24.

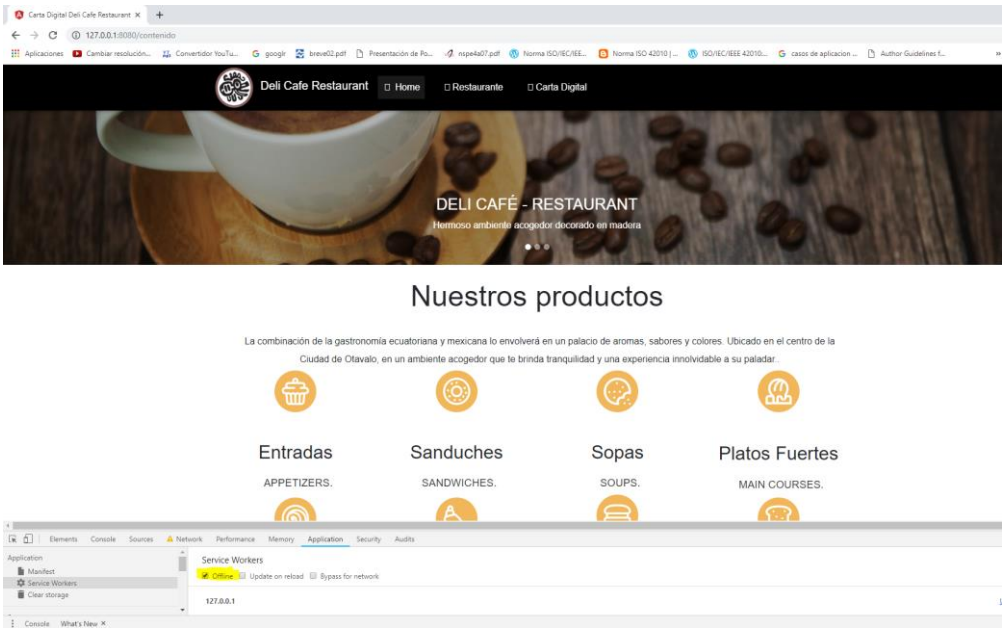



**Fig 24:** Arquitectura App Shell-Funcionalidad

**Fuente:** Propia

**b. Adaptabilidad:** Permitir que la aplicación sea adaptable a cualquier interfaz que puede ser Browser, Web PC y móvil. Para demostrarlo se realiza con la herramienta que ofrece Google Chrome al hacer clic derecho en la web, escoger la opción ***Inspeccionar,*** luego dirigirse a la opción ***Toggle device***, como se muestra en la Figura 25 y escoger el dispositivo.

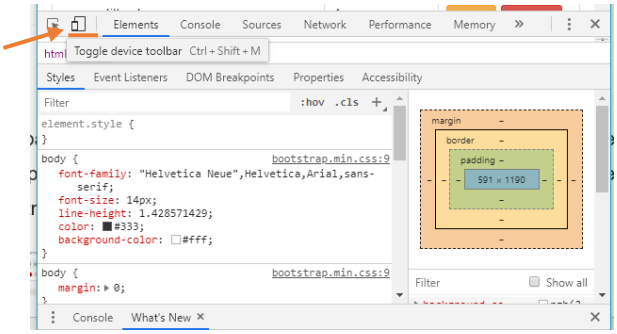



**Fig 25:** Trabajar con cualquier dispositivo

**Fuente:** Propia

- **Responsive Design:** se adapta a cualquier dispositivo como se muestra en la Figura 26.

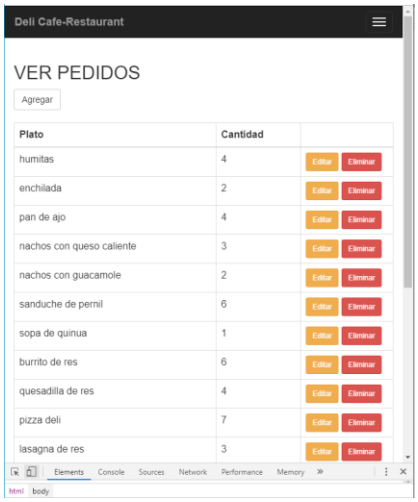



**Fig 26:** Arquitectura Responsive Design-Adaptabilidad-Dispositivo móvil

**Fuente:** Propia

- **SOA:** En la prueba no cumple con el indicador de adaptabilidad como se muestra en la Figura, pero para que pueda cumplir es necesario que el web services sea consumido por otro sistema con interfaz web como se muestra en la Figura 27.

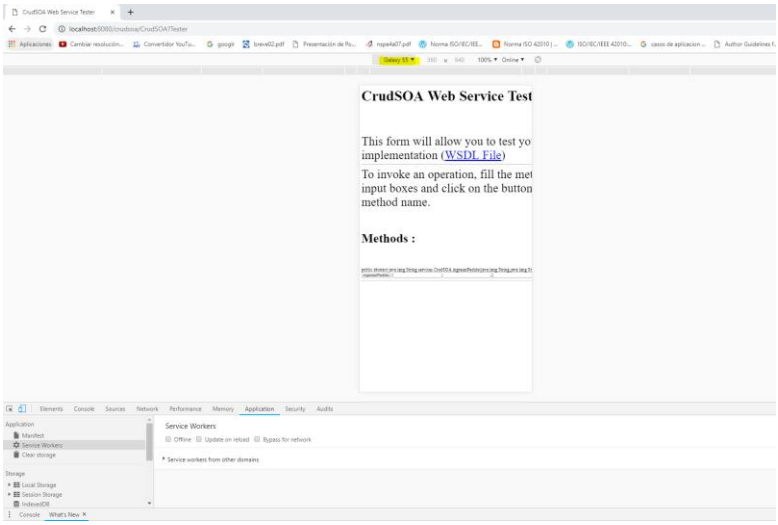



**Fig 27:** Arquitectura SOA-Adaptabilidad-Dispositivo móvil

**Fuente:** Propia

- **App Shell:** Según la prueba si permite la adaptabilidad a cualquier dispositivo como se muestra en la Figura 28.

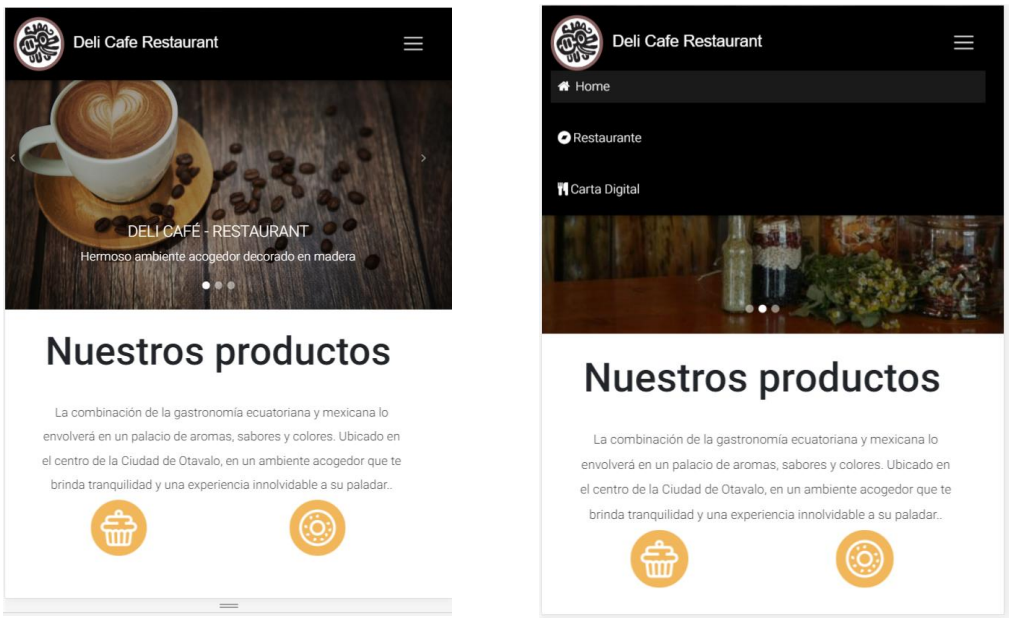



**Fig 28:** Arquitectura App Shell -Adaptabilidad

**Fuente:** Propia

c. **Performance:** Permitir medir el rendimiento de una aplicación en cuanto a respuestas de peticiones, se realiza 20 iteraciones por cada arquitectura, por ende, se optó por la utilización de la herramienta que ofrece Chrome siendo las opciones de Red, Memoria y Rendimiento las que se van a medir.

- **Responsive Design:** Al realizar una petición de ingreso, es decir la primera iteración cuando la cache está vacía se muestra lo siguiente:

✓ **Primera iteración:**

1. **Red:** Al ser la primera iteración se realiza en un tiempo de 138ms como se muestra en la Figura 29.

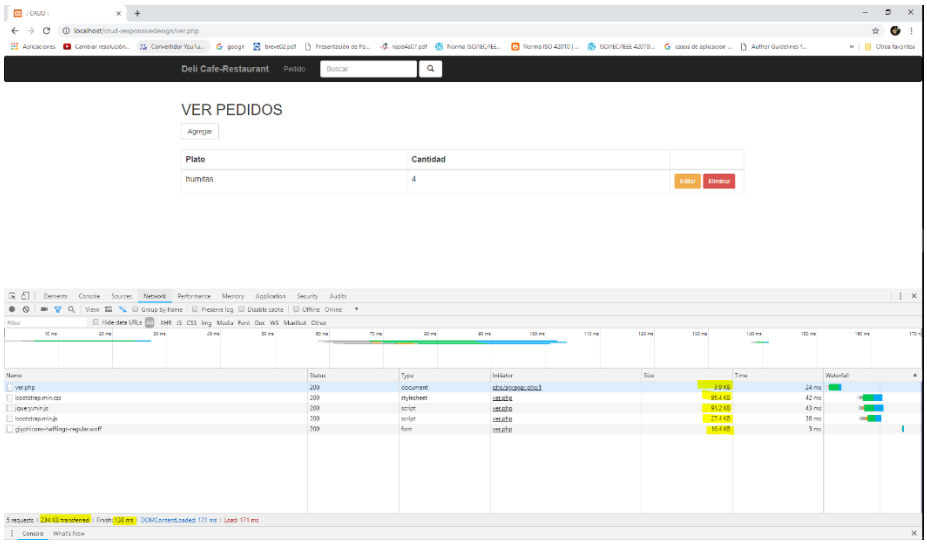



**Fig 29:** Arquitectura Responsive Design-Red

**Fuente:** Propia

2. **Memoria:** Al ser la primera iteración consume 2009 kb como se muestra en la Figura 30.

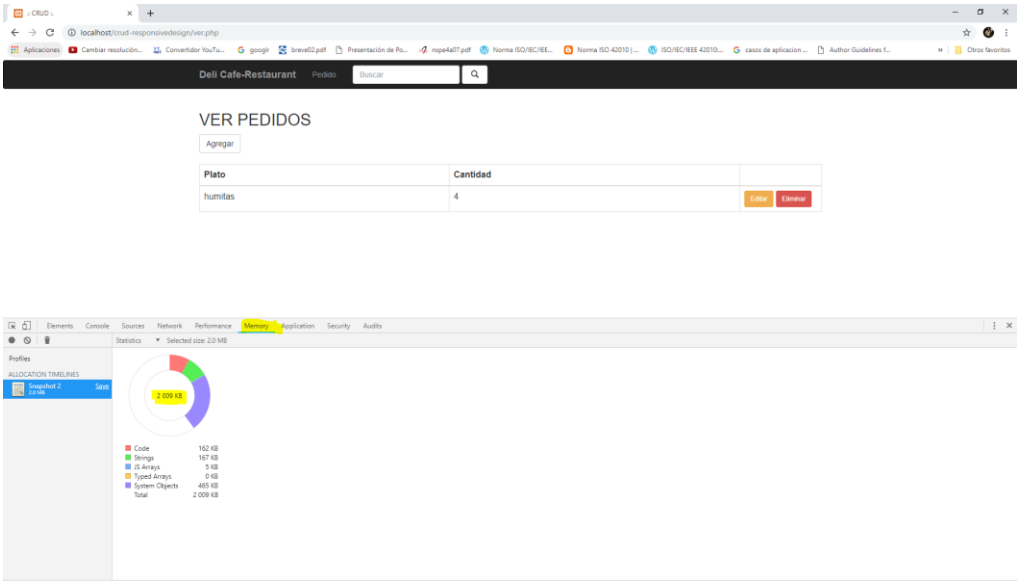



**Fig 30:** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

3. **Rendimiento:** Al ser la primera iteración se realiza en un tiempo 17470ms como se muestra en la Figura 31:

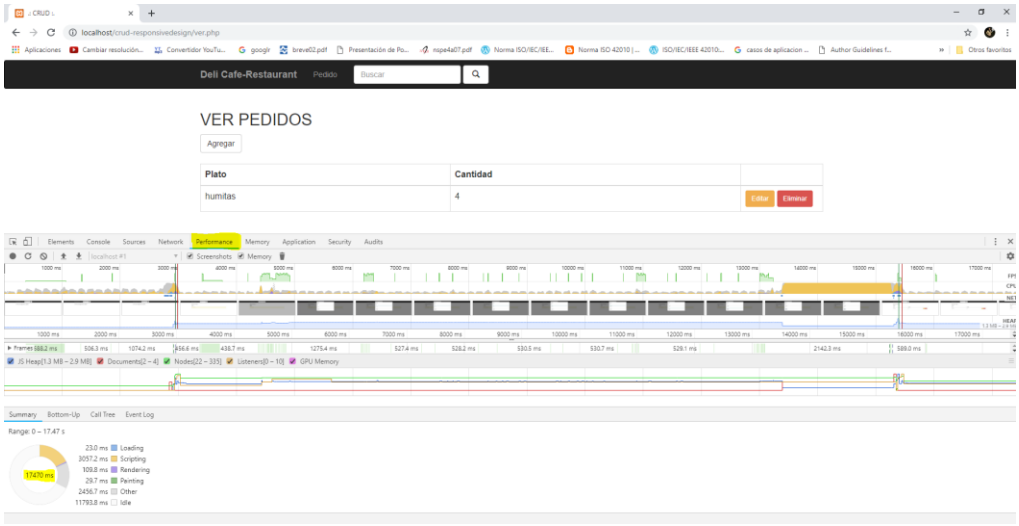



**Fig 31:** Arquitectura Responsive design-rendimiento

**Fuente:** Propia

✓ **Segunda iteración:**

1. **Red:** Al ser la segundad iteración se realiza en un tiempo de 52ms porque ya se encuentra en cache como se muestra en la Figura 32.

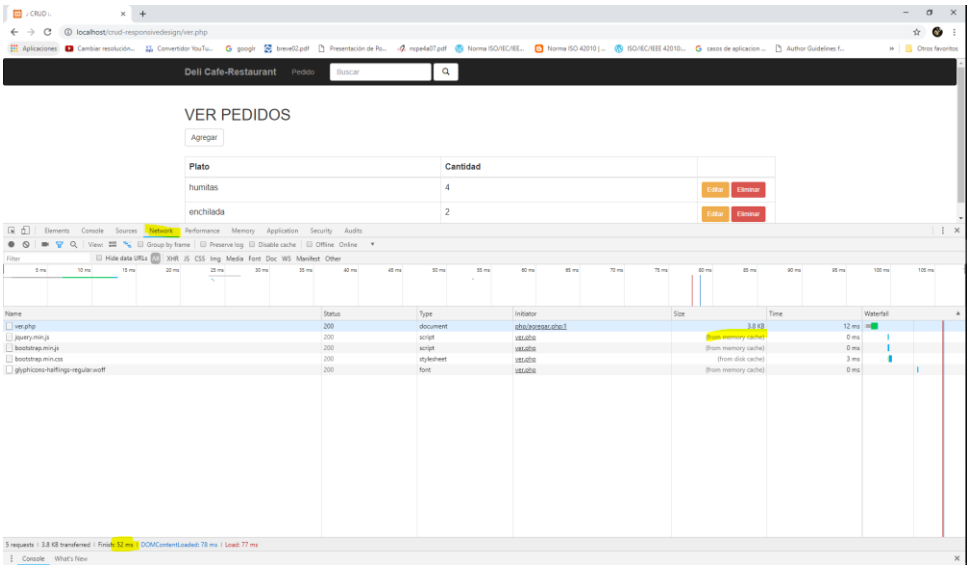



**Fig 32:** Arquitectura Responsive Design-Red

**Fuente:** Propia

2. **Memoria:** Al ser la segunda iteración consume 2012 kb; es decir que aumenta el consumo de memoria como se muestra en la Figura 33.

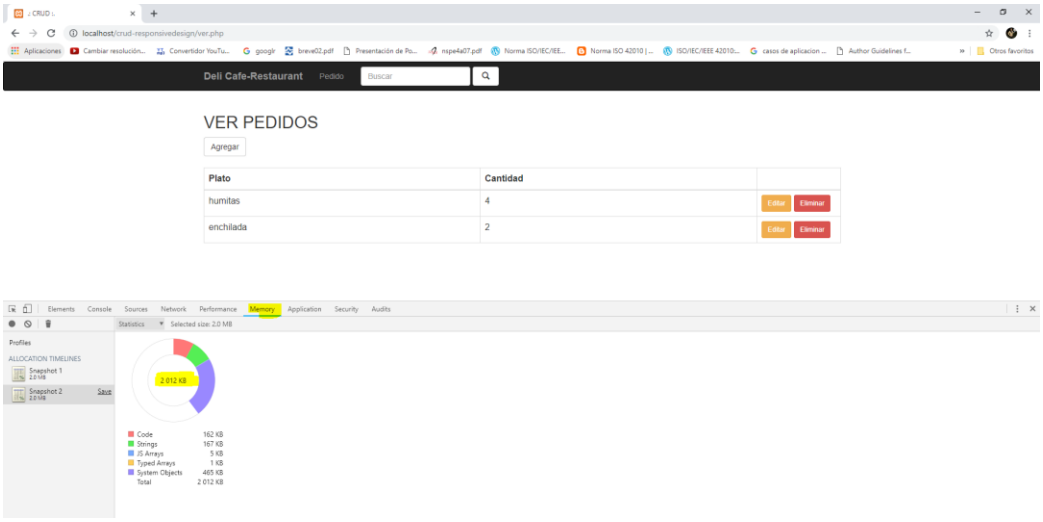



**Fig 33:** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

3. **Rendimiento:** Al ser la segunda iteración se realiza en un tiempo 1103ms; es decir que disminuye el tiempo de rendimiento como se muestra en la Figura 34:

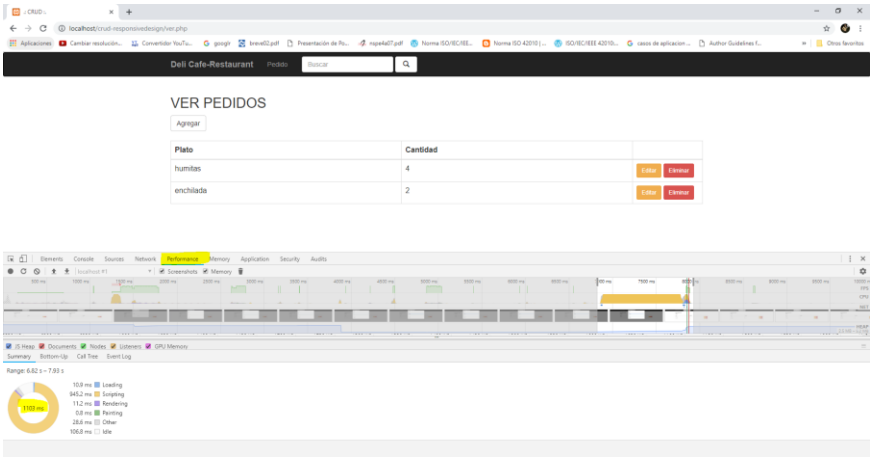



**Fig 34:** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **SOA:** Al realizar una petición de ingreso, es decir la primera iteración cuando la cache está vacía se muestra lo siguiente:

✓ **Primera iteración:**

4. **Red:** Al ser la primera iteración se realiza en un tiempo de 1.03s como se muestra en la Figura 35.

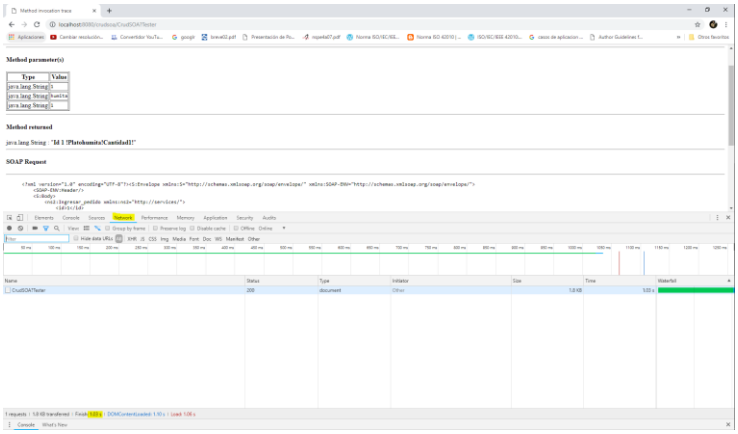


**Fig 35:** Arquitectura SOA-Red

**Fuente:** Propia

5. **Memoria:** Al ser la primera iteración consume 1321 kb como se muestra en la Figura 36.

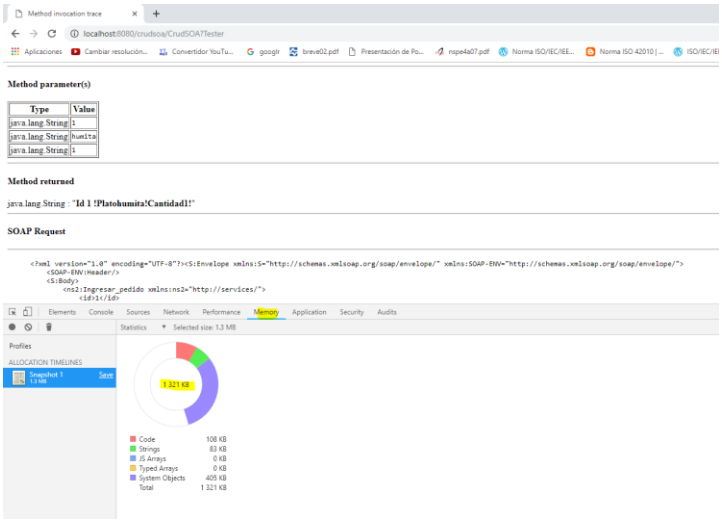



**Fig 36:** Arquitectura SOA-Memoria

**Fuente:** Propia

6. **Rendimiento:** Al ser la primera iteración se realiza en un tiempo 4245ms como se muestra en la Figura 37:

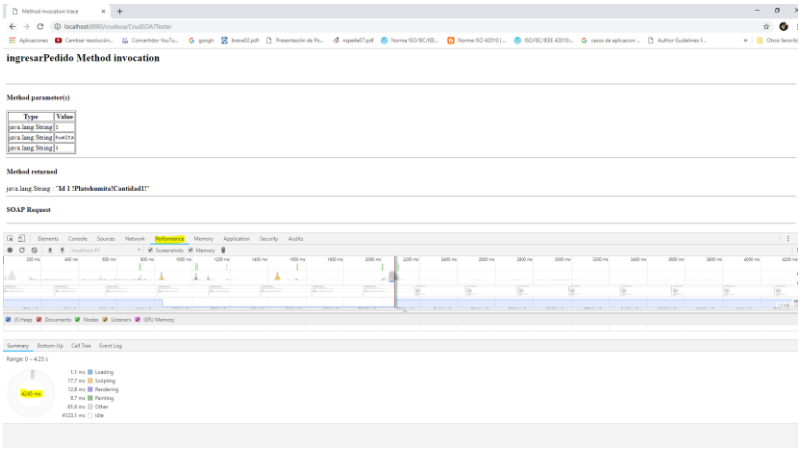



**Fig 37:** Arquitectura SOA-rendimiento

**Fuente:** Propia

✓ **Segunda iteración:**

7. **Red:** Al ser la segunda iteración se realiza en un tiempo de 25ms como se muestra en, no guarda en cache porque solo es un servicio como se muestra en la Figura 38.

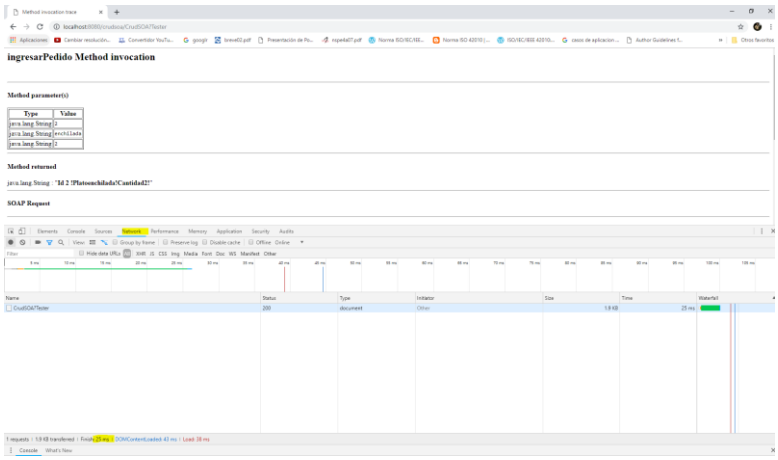



**Fig 38:** Arquitectura SOA-Red

**Fuente:** Propia

8. **Memoria:** Al ser la segunda iteración consume 1325kb como se muestra en la Figura 39.

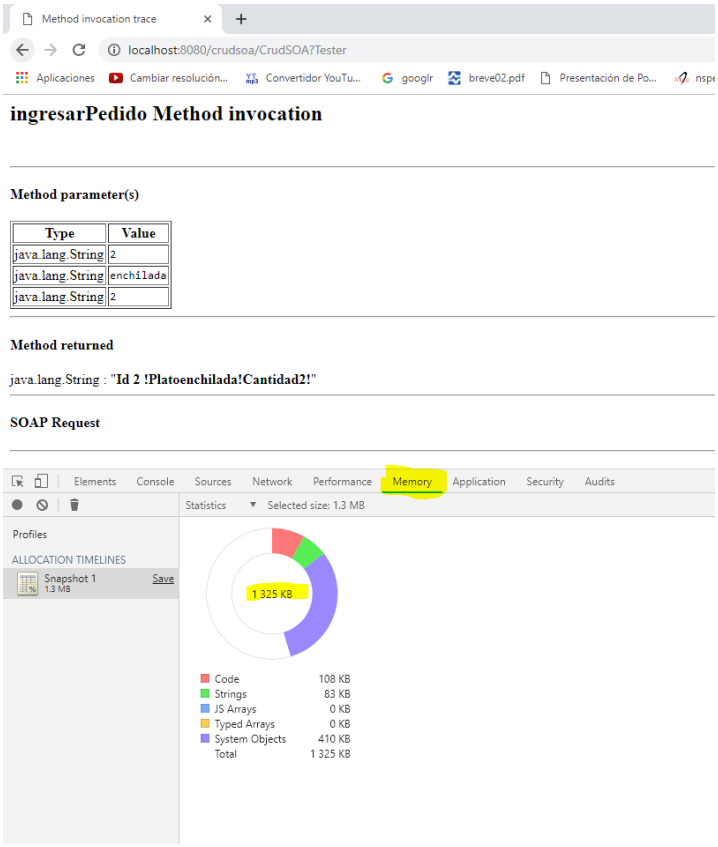



**Fig 39:** Arquitectura SOA-Memoria

**Fuente:** Propia

9. **Rendimiento:** Al ser la segunda iteración se realiza en un tiempo 4130ms como se muestra en la Figura 40:

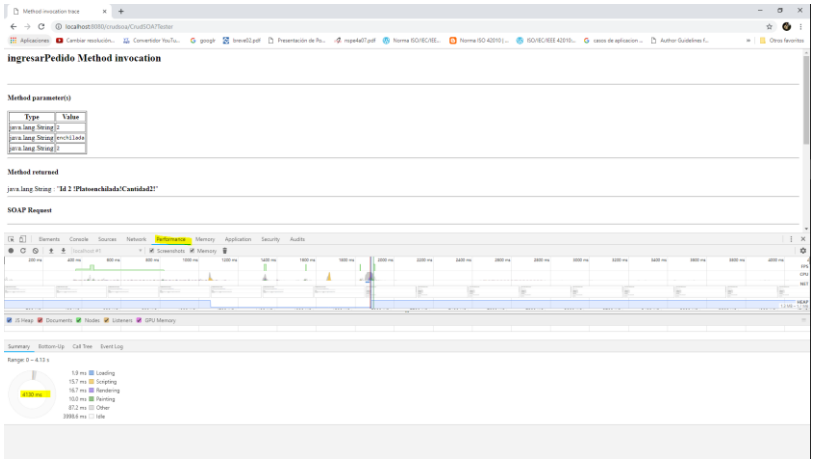


**Fig 40:** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **App Shell:** Al realizar una petición de ingreso, es decir la primera iteración cuando la cache está vacía se muestra lo siguiente:

✓ **Primera iteración:**

10. **Red:** Al ser la primera iteración se realiza en un tiempo de 7.60s como se muestra en la Figura 41.

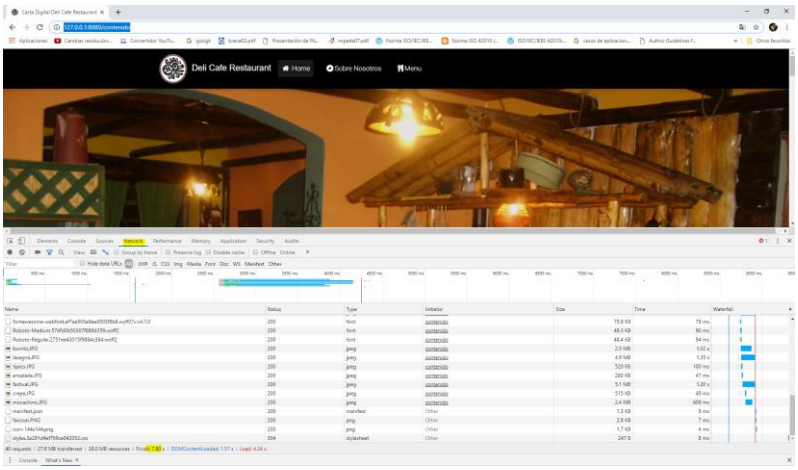



**Fig 41:** Arquitectura App Shell-Red

**Fuente:** Propia

11. **Memoria:** Al ser la primera iteración consume 8664 kb como se muestra en la Figura 42.

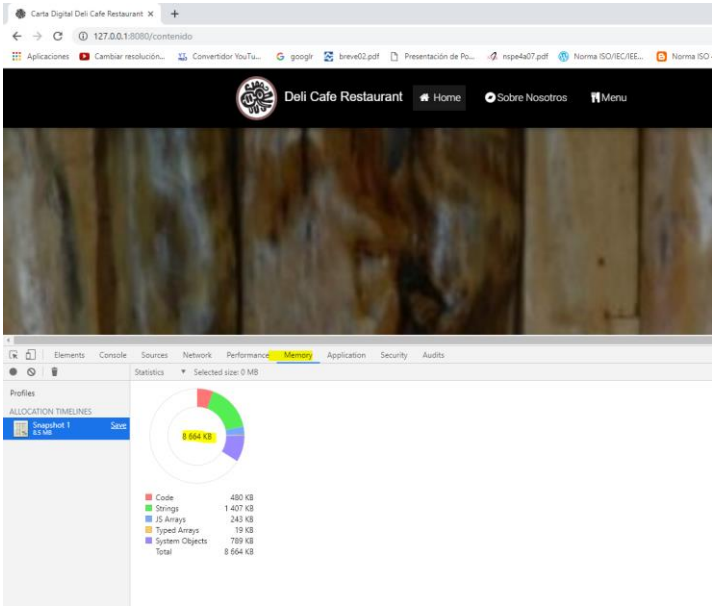



**Fig 42:** Arquitectura App Shell-Memoria

**Fuente:** Propia

12. **Rendimiento:** Al ser la primera iteración se realiza en un tiempo 7856ms como se muestra en la Figura 43:

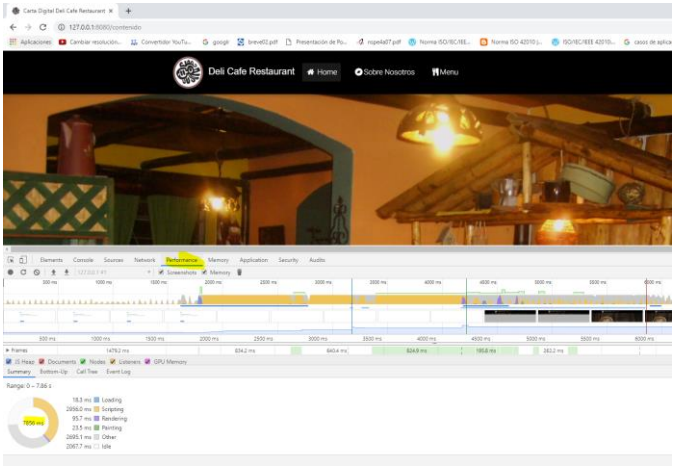



**Fig 43:** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **App Shell:** Al realizar una petición de ingreso, es decir la segunda iteración cuando la cache ya está llena se muestra lo siguiente:

✓ **Segunda iteración:**

13. **Red:** Al ser la segunda iteración se realiza en un tiempo de 7.54s como se muestra en la Figura 44.

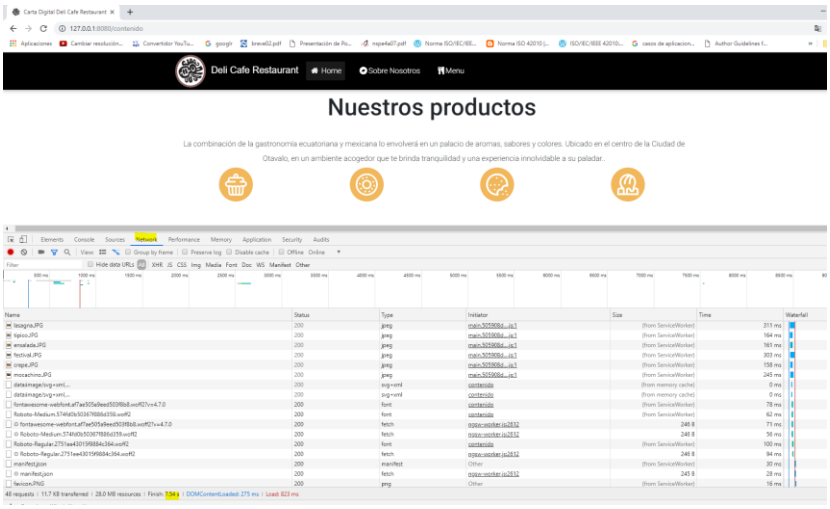



**Fig 44:** Arquitectura App Shell-Red

**Fuente:** Propia

14. **Memoria:** Al ser la segunda iteración consume 8656kb como se muestra en la Figura 45.

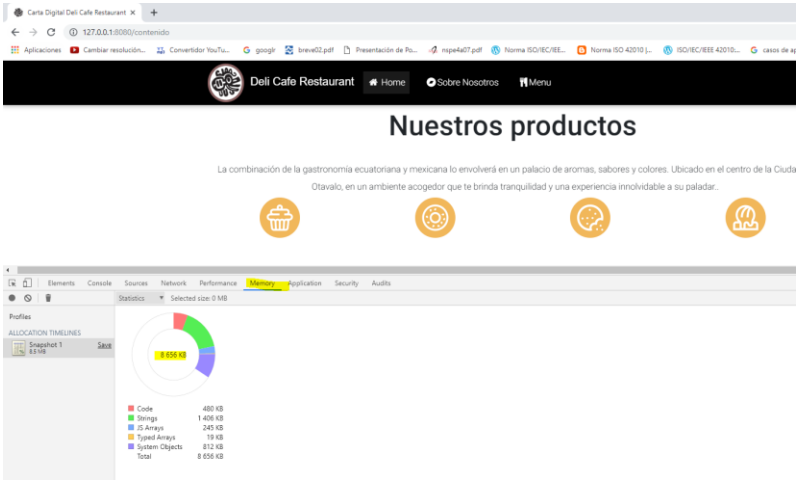



**Fig 45:** Arquitectura App Shell-Memoria
**Fuente:** Propia

15. **Rendimiento:** Al ser la segunda iteración se realiza en un tiempo 1036ms como se muestra en la Figura 46:

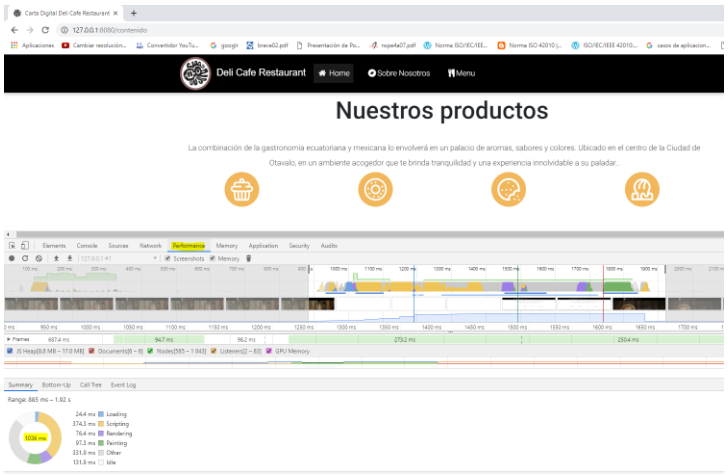



**Fig 46:** Arquitectura App Shell-rendimiento
**Fuente:** Propia

Ahora al tener realizado los laboratorios para el indicador de ***Performance*** de cada arquitectura se muestra los siguientes datos:

**Red:** De acuerdo a cada iteración se muestra el tiempo de carga de peticiones como se muestra en la Tabla 17 y en la Figura 47.

| RED/Milisegundos | | | |
|---|---|---|---|
| Iteraciones | Responsive Design | SOA | App Shell |
| I1 | 138 ms | 1030 ms | 7600 ms |
| 2 | 52 ms | 25 ms | 7540 ms |
| 3 | 116 ms | 38 ms | 1050 ms |
| 4 | 117 ms | 23 ms | 1200 ms |
| 5 | 102 ms | 25 ms | 1010 ms |
| 6 | 49 ms | 24 ms | 1150 ms |
| 7 | 47 ms | 20 ms | 1170 ms |
| 8 | 43 ms | 19 ms | 1280 ms |
| 9 | 47 ms | 22 ms | 566 ms |
| 10 | 71 ms | 28 ms | 756 ms |
| 11 | 50 ms | 22 ms | 689 ms |
| 12 | 78 ms | 22 ms | 793 ms |
| 13 | 45 ms | 14 ms | 585 ms |
| 14 | 70 ms | 17 ms | 761 ms |
| 15 | 76 ms | 24 ms | 761 ms |
| 16 | 71 ms | 16 ms | 718 ms |
| 17 | 64 ms | 15 ms | 566 ms |
| 18 | 51 ms | 18 ms | 718 ms |
| 19 | 101 ms | 15 ms | 699 ms |
| 20 | 73 ms | 23 ms | 694 ms |

**Tabla 17:** Iteraciones-RED

**Fuente:** Propia

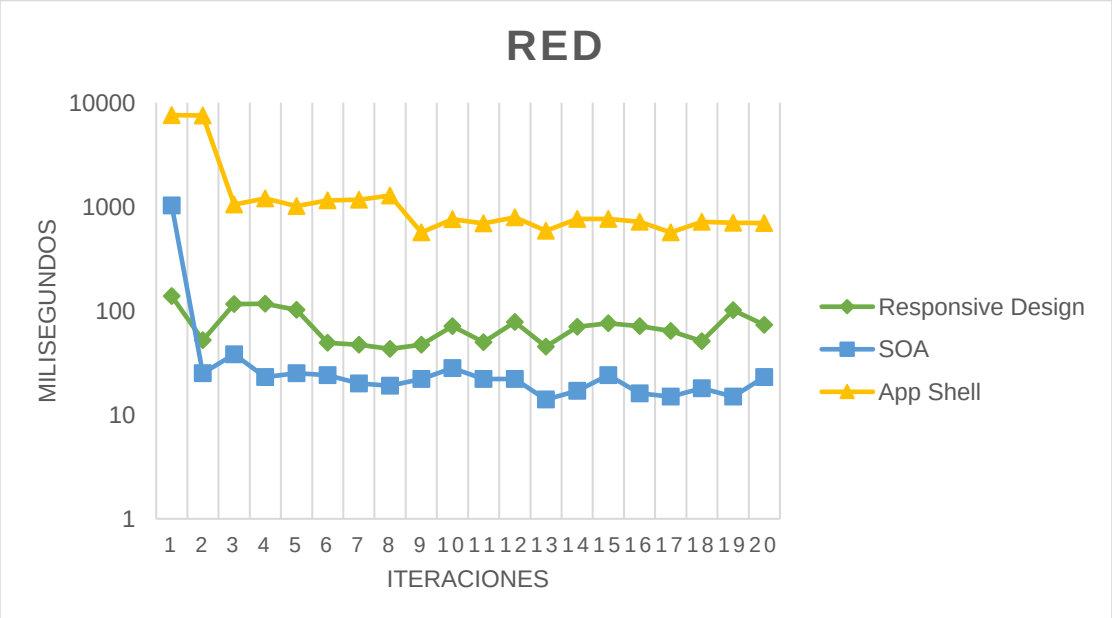



**Fig 47:** Indicador RED

**Fuente:** Propia

De acuerdo a la Figura 47, se observa en la gráfica referente a la Arquitectura App Shell( representada por la franja naranja), que en las dos primeras iteraciones al ejecutar la aplicación, el tiempo de carga se encuentra en un rango de 7600ms a 7540ms porque esta almacenando en la cache cada elemento de la aplicación implicando así su demora, posteriormente a partir de la tercera iteración muestra una disminución notable al volver a ejecutar la aplicación, debido a que, cada elemento ya está guardada en la cache, permitiendo mantenerse en un rango de 1050ms a 694ms.

Por otro lado, la Arquitectura SOA (representada por la franja azul) en la primera iteración alcanza un tiempo de carga de 1030ms, a partir de la segunda iteración se encuentra en un rango de 25ms a 23ms siendo así la Arquitectura que menos tiempo de carga tiene en cuanto al indicador Red siendo hasta el momento la más aceptable.

Por último, la Arquitectura Responsive Design (representada por la franja verde), en la primera iteración alcanza un tiempo de carga de 138ms, a partir de la segunda iteración 52ms a 73ms.

De acuerdo al análisis realizado se concluye que, en la medición de este indicador, la más idónea en cuanto al tiempo de carga de la aplicación resulta ser SOA.

**Memoria:** De acuerdo al análisis de cada iteración se muestra el espacio que almacenando la aplicación de acuerdo a cada petición como se observar muestra en la Tabla18 y en la Figura 48.

| MEMORIA/KB | | | |
|---|---|---|---|
| Iteraciones | Responsive Design | SOA | App Shell |
| 1 | 2009 KB | 1321 KB | 8664 KB |
| 2 | 2012 KB | 1325 KB | 8656 KB |
| 3 | 2595 KB | 1853 KB | 8691 KB |
| 4 | 2487 KB | 1830 KB | 1608 KB |
| 5 | 2498 KB | 1307 KB | 1634 KB |
| 6 | 2293 KB | 1311 KB | 1655 KB |
| 7 | 2685 KB | 1311 KB | 1671 KB |
| 8 | 2335 KB | 1317 KB | 1644 KB |
| 9 | 2336 KB | 1317 KB | 1675 KB |
| 10 | 2714 KB | 1326 KB | 1675 KB |
| 11 | 2717 KB | 1326 KB | 1675 KB |
| 12 | 1252 KB | 1326 KB | 1675 KB |
| 13 | 1252 KB | 1326 KB | 1675 KB |
| 14 | 2382 KB | 1326 KB | 1730 KB |
| 15 | 2383 KB | 1340 KB | 1734 KB |
| 16 | 2387 KB | 1340 KB | 1734 KB |

| | | | |
|---|---|---|---|
| 17 | 2389 KB | 1847 KB | 1634 KB |
| 18 | 2390 KB | 1847 KB | 1634 KB |
| 19 | 2493 KB | 1854 KB | 1634 KB |
| 20 | 2423 KB | 1854 KB | 1634 KB |

**Tabla 18:** Iteraciones-Memoria

**Fuente:** Propia

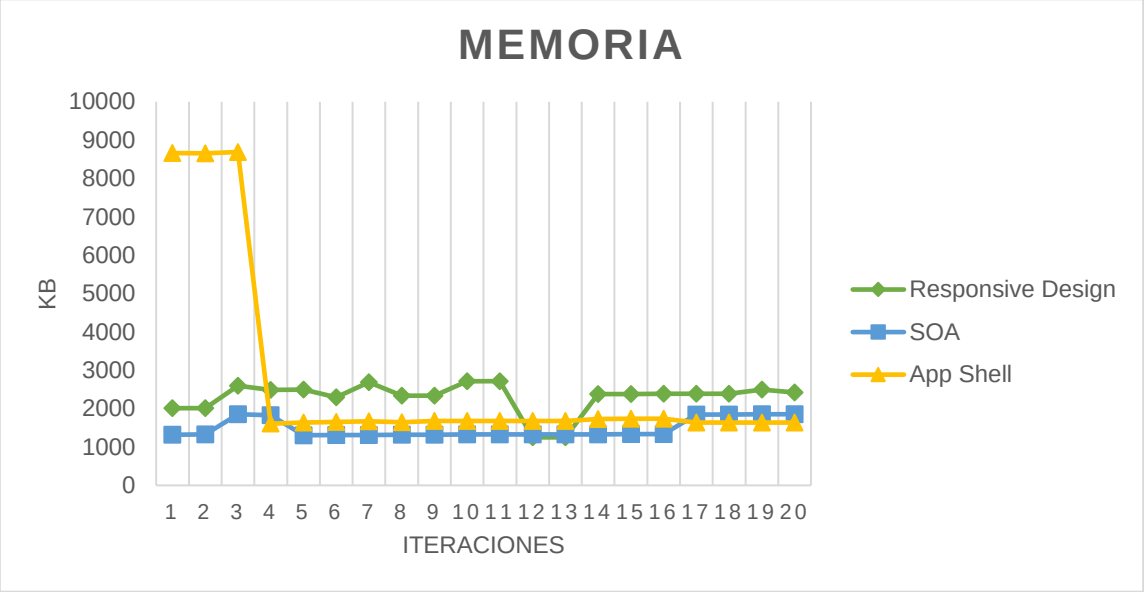



**Fig 48:** Indicador Memoria

**Fuente:** Propia

De acuerdo a la Figura 48, se observa en la gráfica referente a la Arquitectura App Shell( representada por la franja naranja), que en las tres primeras iteraciones al ejecutar la aplicación, el espacio que ocupa se encuentra en un rango de 8664kb a 8691kb porque esta almacenando en la cache cada elemento de la aplicación produciendo así un consumo mayor en recursos de memoria, posteriormente a partir de la cuarta iteración muestra una disminución en el consumo de memoria , debido a que, cada elemento al estar guardada en la cache tiende a ser más ligera en el consumo de memoria, permitiendo mantenerse en un rango de 1608kb a 1634kb, siendo así la Arquitectura que menos memoria utiliza al ejecutar la aplicación siendo hasta el momento la más aceptable.

Por otro lado, la Arquitectura SOA (representada por la franja azul) en las dos primeras iteraciones se encuentra en un rango de 1321kb a 135kb que ocupa de memoria, pero a partir de la tercera iteración se encuentra en un rango de 1853kb a 1854kb, utilizando así más memoria.

Por último, la Arquitectura Responsive Design (representada por la franja verde) en las dos primeras iteraciones se encuentra en un rango de 2009kb a 2012kb que ocupa de memoria, pero a partir de la tercera iteración se encuentra en un rango de 2595kb a 2423kb, utilizando así más memoria.

De acuerdo al análisis realizado se concluye que, en la medición de este indicador, la más idónea en cuanto al tiempo de respuesta resulta ser App Shell.

**Rendimiento:** De acuerdo a cada iteración se muestra el tiempo de respuesta de las peticiones como se muestra en la Tabla19 y en la Figura 49.

| Rendimiento/ms | | | |
|---|---|---|---|
| Iteraciones | Responsive Design | SOA | App Shell |
| 1 | 17470 ms | 4245 ms | 7856 ms |
| 2 | 1103 ms | 4130 ms | 1036 ms |
| 3 | 2415 ms | 3423 ms | 1073 ms |
| 4 | 2259 ms | 4373 ms | 1249 ms |
| 5 | 9353 ms | 5897 ms | 1171 ms |
| 6 | 1932 ms | 4092 ms | 1062 ms |
| 7 | 1643 ms | 5253 ms | 1090 ms |
| 8 | 1448 ms | 6304 ms | 699 ms |
| 9 | 1213 ms | 6304 ms | 758 ms |
| 10 | 1037 ms | 8274 ms | 936 ms |
| 11 | 2019 ms | 7944 ms | 793 ms |
| 12 | 2358 ms | 8266 ms | 836 ms |
| 13 | 1702 ms | 7515 ms | 789 ms |
| 14 | 2226 ms | 7530 ms | 744 ms |
| 15 | 1106 ms | 6504 ms | 802 ms |
| 16 | 1108 ms | 7626 ms | 325 ms |
| 17 | 1646 ms | 7927 ms | 325 ms |
| 18 | 1146 ms | 7771 ms | 325 ms |
| 19 | 1192 ms | 7242 ms | 325 ms |
| 20 | 1192 ms | 7235 ms | 325 ms |

**Tabla 19:** Iteraciones-Rendimiento

**Fuente:** Propia

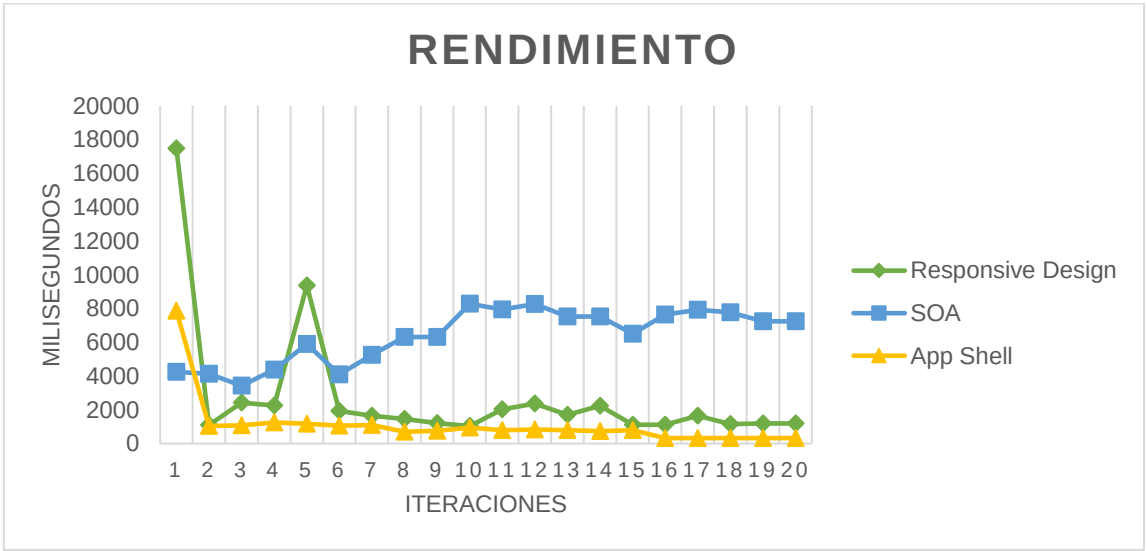



**Fig 49:** Indicador Rendimiento

**Fuente:** Propia

De acuerdo a la Figura 49, se observa en la gráfica referente a la Arquitectura App Shell( representada por la franja naranja), que en la primera iteración al ejecutar la aplicación, el tiempo de respuesta a una petición es de 7856ms, porque esta almacenando en la cache cada petición que se va realizando, posteriormente a partir de la segunda iteración el tiempo de respuesta a una petición disminuye debido a que, todos los elementos se encuentran almacenados en cache, permitiendo mantenerse en un rango de 1036ms a 325ms, siendo así la Arquitectura que menos se demora en responder una petición, logrando ser hasta el momento la más aceptable.

Por otro lado, la Arquitectura SOA (representada por la franja azul) en las dos primeras iteraciones se encuentra en un rango de 4245ms a 4130ms en responder una petición, pero a partir de la tercera iteración se encuentra en un rango de 3423ms a 7235, siendo así la que más se demora en responder una petición.

Por último, la Arquitectura Responsive Design (representada por la franja verde) en la primera iteración alcanza un tiempo de respuesta de 17470ms, a partir de la segunda iteración se encuentra entre un rango de 1103ms a 1192ms.

De acuerdo al análisis realizado se concluye que, la más idónea en cuanto al rendimiento de la aplicación resulta ser App Shell.

### 2.4.2. Técnica de Evaluación

En la evaluación de las arquitecturas se utiliza la norma ISO 42010 como punto de partida; y en conjunto con sus modelos se determina varios indicadores medibles a fin de tener un análisis más detallado en referencia a la arquitectura que resulte ser apropiada en la construcción de Web Progresivas.

Se definió ciertos parámetros como se detalla a continuación a partir de la técnica de evaluación de QFD en conjunto con la escala de Likert donde se estimula un valor mínimo como el resultado más desfavorable y máximo como el más favorable (Díaz, 2019):

| Parámetro | Valor |
|---|---|
| **No cumple** | 0 |
| **Cumple medianamente** | 3 |
| **Cumple en su totalidad** | 5 |

**Tabla 20:** Parámetros

**Fuente:** Propia

Partiendo de ello, se procede a dar el valor que corresponda a cada arquitectura de acuerdo a las pruebas de cada indicador como se muestra en la Tabla 21.

| | Responsive Design | SOA | APP SHELL |
|---|---|---|---|
| Funcionalidad | 0 | 0 | 5 |
| Adaptabilidad | 5 | 0 | 5 |
| Red | 3 | 5 | 0 |
| Memoria | 0 | 3 | 5 |
| Rendimiento | 3 | 0 | 5 |
| Total | 11 | 8 | 20 |
| Porcentaje % | 44 | 32 | 80 |

**Tabla 21:** Indicadores-Medición

**Fuente:** Propia

A continuación, se muestra las imágenes determinando la Arquitectura idónea:

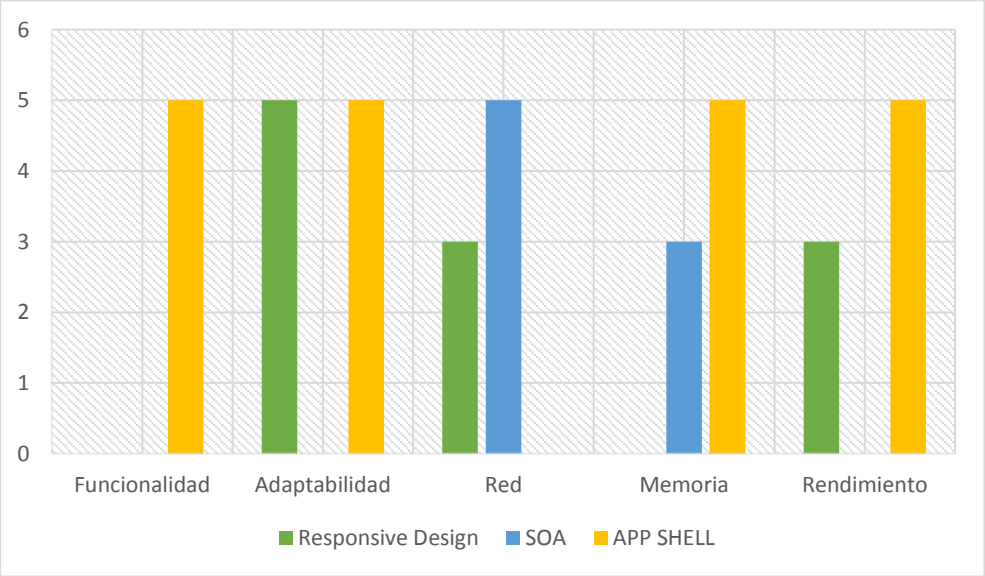



**Fig 50:** Puntuación de cada indicador

**Fuente:** Propia

En la Figura 51 se muestra el porcentaje de cada arquitectura obtenida de la prueba de cada indicador dando como ganador a App Shell con un 80%. Es decir que posee casi en su totalidad con los parámetros necesarios para la construcción de una Web progresiva.

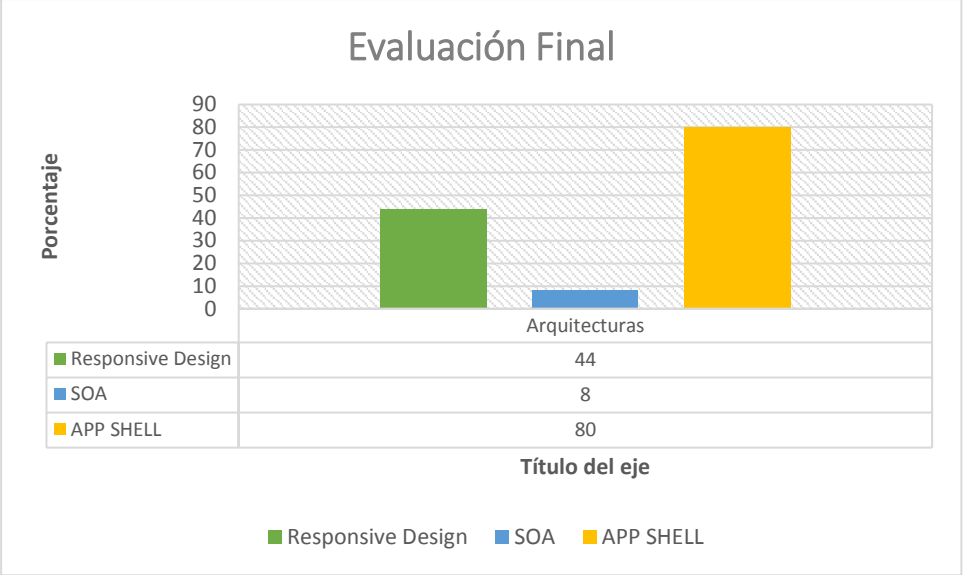



**Fig 51:** Análisis en porcentaje de las Arquitecturas

**Fuente:** Propia

Al tener ya la arquitectura selecciona en esta etapa para realizar la documentación se usarán estilos arquitectónicos y la selección y priorización se llevara a través de los siguientes pasos (Maceda & Careaga, 2016):

### 2.4.3. ATAM

Para la evaluación de la arquitectura escogida como lo es App Shell se opta por un método llamado ATAM que permite evaluar una arquitectura, a través de las decisiones arquitectónicas en relación a determinados atributos de calidad (Valencia, 2016). A continuación, se presenta las características del método:

- ATAM ayuda a predecir como un atributo de interés puede ser afectado por decisiones arquitectónicas.
- Los atributos de calidad de interés son clarificados mediante el análisis de los escenarios definidos por los distintos stakeholders en términos de estímulos y respuestas.
- Una vez los escenarios han sido definidos y priorizados el siguiente paso es definir que enfoques arquitectónicos pueden afectar a esos atributos de calidad.

### 2.4.4. Fases de ATAM

**FASE 1: Presentación**

- Paso 1. Presentar el método ATAM: se presenta el método a los evaluadores y stakeholders.
- Paso 2. Presentar los objetivos de negocio o drivers de negocio: Se presentan las metas de negocio que motivan el esfuerzo de desarrollo y que deberán ser brindados por la arquitectura.
- Paso 3. Presentar la arquitectura: Presentar la arquitectura haciendo énfasis en como aborda los objetivos de negocio.

**FASE 2: Investigación y análisis**

- Paso 4. Identificar de las decisiones arquitectónicos: los arquitectos presentan los enfoques arquitectónicos para este dominio / arquitectura para hacer frente a las metas del sistema y los objetivos de negocio.
- Paso 5. Generar el árbol de utilidad: Los factores de calidad (fiabilidad, disponibilidad, rendimiento) son identificados y especificados hasta el nivel de escenarios, con estímulos y respuestas y finalmente priorizados.
- Paso 6. Analizar los enfoques arquitectónicos: los stakeholders y el arquitecto analizan como los enfoques arquitectónicos afectan a los factores identificados en el paso anterior.

**FASE 3: Testing**

- Paso 7. Definir Riesgo y no riesgos

### 2.4.5. Aplicación del método

Una vez definido las fases del método ATAM se procede a aplicarlos en la investigación.

**FASE 1:**

- Paso 1. Presentación de ATAM

En el caso de estudio, el método fue presentado al tutor de la aplicación y por ende al desarrollador que son los stakeholders como se muestra en Tabla 12.

- Paso 2. Presentación de los objetivos de negocio:

Los objetivos de negocio ya están previamente establecidos como se muestra en la Tabla 3: Objetivos de negocio.

- Paso 3. Presentación de la arquitectura:

Para la aplicación se opta por la Arquitectura App Shell que trabaja en conjunto con el patrón MVVM, la App Shell está enfocada en almacenar la Shell en cache utilizando un service worker sin utilizar una red como se muestra en la Figura 14.

**FASE 2:**

- Paso 4. Identificar las decisiones arquitectónicas

Las decisiones arquitectónicas más relevantes tomadas de acuerdo al caso de estudio son:

Se estructura la aplicación en capas con la utilización Shell, estableciendo un determinado número de elementos correspondientes al dominio del problema. Por ello es necesario el uso del Framework Angular V6. A este tipo de estructura se incorpora el modelo el modelo MVVM que permite realizar un trabajo enfocado a la Interfaz de usuario.

- Paso5. Generar el árbol de utilidad

Es necesario tomar en cuenta que este árbol de utilidad se lo utiliza cuando es reducido el número de personas participando en la evaluación, como lo es en este caso de estudio. En primer ámbito se inicia con la utilización de los drivers identificados anteriormente, para poder plantear posibles escenarios que lo cubran.

En la Tabla 22 está presentado la funcionalidad principal del driver de la aplicación, en conjunto con los requerimientos de atributos de calidad como lo es el desempeño, disponibilidad del sistema y usabilidad. A continuación, se muestra el árbol de utilidad establecido.

| | | |
|---|---|---|
| Utilidad | Funcionalidad | Básica |
| | Interoperabilidad | Sistemas legados |
| | Desempeño | Tiempo de respuesta |
| | Disponibilidad | Fallas de software |
| | | Fallas de hardware |
| | | Fallas de red |

| | | |
|---|---|---|
| | Usabilidad | Interfaz web |
| | | Interfaz en dispositivos móviles |

**Tabla 22:** Árbol de utilidad

**Fuente:** Propia

Como el objetivo es encontrar escenarios para detectar riesgos en la posible arquitectura hay que desglosar el árbol de utilidad.

| | | | |
|---|---|---|---|
| Utilidad | Funcionalidad | Básica | Permitir interactuar con la aplicación aunque no haya conexión . |
| | | Deseable | Básica |
| | Interoperabilidad | Sistemas legados(futuro-facturación electrónica) | En un futuro implementación de una base de datos para volverla transaccional. |
| | Disponibilidad | Fallas de software | Falla de la BDD-cache |
| | | Fallas de hardware | Fallas del servidor |
| | | Fallas de red | Fallas del archivo que ejecuta offline |
| | Usabilidad | Interfaz web | Acceso a la aplicación desde la interfaz web |
| | | Interfaz en dispositivos móviles | Acceso a la aplicación desde dispositivos |

**Tabla 23:** Árbol de utilidad-desglosada

**Fuente:** Propia

Ahora se muestra la relevancia de cada escenario para el negocio.

- Paso 6. Análisis de las decisiones arquitectónica

Para realizar el proceso del análisis de las decisiones se selecciona los escenarios primordiales para el negocio, además es posible combinar algunos de los escenarios para la simplificación del análisis. A continuación, se muestra el escenario principal a ser utilizado:

a) Funcionamiento de la aplicación en un estado Offline.

Para el escenario estipulado se realizará el análisis a través del análisis de enfoques arquitectónicos como se muestra en la Tabla 24.

| Análisis de Enfoques Arquitectónicos | |
|---|---|
| Escenario nro1 | Utilizar la aplicación si necesidad de red. |
| Atributos | Funcionalidad y usabilidad |
| Entorno | Un cliente ingresa a la aplicación desde cualquier otro dispositivo |
| Estímulo | Solicitud de funcionamiento desde la cache. |
| Respuesta | Funcionalidad: Trabaja en segundo plano de manera correcta. |

| | | | | |
|---|---|---|---|---|
| | Usabilidad: el usuario realiza de manera intuitiva la operación desde el navegador. | | | |
| Decisiones arquitectónicas | Punto de sensibilidad | Equilibrio | Riesgo | No-Riesgo |
| Patron MVVM, framework Angular | Con el diseño el punto de sensibilidades de una transacción a la vez pues no hay indicaciones de concurrencia simultaneas. | Existe un equilibrio entre mantenibilidad y desempeño pues el modelo de la arquitectura App Shell lo facilita. | R1: Durante la utilización de la aplicación no responda a la brevedad cuando no haya conexión a la red | NR1: Mantenibilidad de las capas por su separación. NR2: Integración de capas entre sí. NR3: Usabilidad por medio de la interfaz web o no web. |
| Dentro de la vista modelo se encuentra la capa de presentación | Una transacción simultánea bajo el argumento del punto de sensibilidad de la decisión arquitectónica anterior | Facilidad de uso y desempeño pues la capa presentación dentro de la vista modelo está involucrada el modelo y dentro de ella se encuentra las otras capas lo que incrementa el número de elementos procesando las transacciones y de esta manera afectando al desempeño . | R2: No hay evidencia de que la arquitectura haga manejo de transacciones simultaneas, es decir que sea transaccional. | NR1: Mantenibilidad de las capas por su separación. NR2: Integración de capas entre sí. NR3: Usabilidad por media de la interfaz web o no web. |
| Razonamiento | Este escenario es primordial y las decisiones arquitectónicas planteadas pueden ser implementadas. Sin embargo, la falta de detalles genera riesgo por lo que se debe asegurarse que la arquitectura lo cumpla. | | | |

**Tabla 24:** Análisis del escenario

**Fuente:** Propia

**FASE 3: Testing**

- Paso 7. Resultados de la evaluación

Al analizar los escenarios propuestos para la evaluación, se ha identificado los siguientes resultados:

**Riesgos:**

R1: Durante la utilización de la aplicación no responda a la brevedad cuando no haya conexión a la red.

R2: No hay evidencia de que la arquitectura haga manejo de transacciones simultaneas, es decir que sea transaccional.

**No-Riesgos**

NR1: Mantenibilidad de las capas por su separación.

NR2: Integración de capas entre sí.

NR3: Usabilidad por media de la interfaz web o no web.

## 2.5. Herramientas de desarrollo

A continuación, se detalla las herramientas a utilizar para el desarrollo de la aplicación Web Progresiva de la gestión de pedidos del Café- Restaurant "Deli- Café Restaurant"

### 2.1.1. Angular Versión 6

Angular es JavaScript. Es un proyecto de código abierto, realizado en JavaScript que contiene un conjunto de librerías útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo (Robles, 2018).

JavaScript con programación del lado del cliente.

Es decir, es un framework de desarrollo para JavaScript creado por Google. Con la finalidad de facilitar el desarrollo de aplicaciones web SPA y además proveer herramientas para trabajar con los elementos de una web de una manera más sencilla y óptima.

### 2.1.2. Material Design Bootstrap

Es una librería que trabaja con el material design y bootstrap 4 para permitir el desarrollo de una interfaz más amigables y adaptables a celulares y tablets de modo responsivo.

# CAPÍTULO 3

## Resultados

Para la validación de los resultados, se realizó a través de los valores obtenidos de la encuesta ver Anexo 3, que fue realizada con la herramienta para la creación de formularios online mediante el uso de Google Forms, la recolección de datos fue realizada con el intérprete de resultados preliminares de la misma herramienta, la curación y validación de datos se aplicó el Método de Chi cuadrado para la validación de la hipótesis planteada en la investigación.

A continuación, se muestra en la Tabla 25 los resultados por pregunta de la encuesta realizada a 27 personas.

| Pregunta | SI | NO |
|---|---|---|
| Pregunta 1 | 12 | 15 |
| Pregunta 2 | 19 | 8 |
| Pregunta 3 | 16 | 11 |
| Pregunta 4 | 18 | 9 |
| Pregunta 5 | 13 | 14 |
| Pregunta 6 | 23 | 4 |
| Pregunta 7 | 24 | 3 |
| Pregunta 8 | 24 | 3 |
| Pregunta 10 | 25 | 2 |

| | Todas las anteriores | opción a | opción b | opción b |
|---|---|---|---|---|
| Pregunta 9 | 13 | 3 | 4 | 7 |

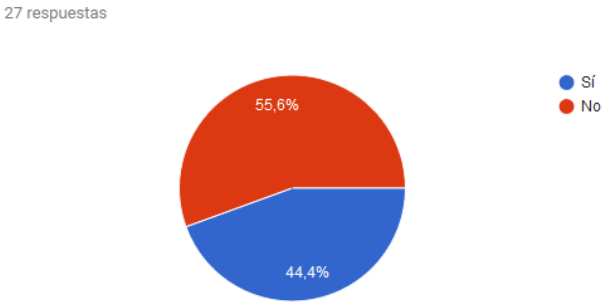
**Tabla 25:** Resultados encuesta

**Fuente:** Propia

Además, se detalla las gráficas de los resultados por cada pregunta de la encuesta realizada.

**Pregunta 1:**

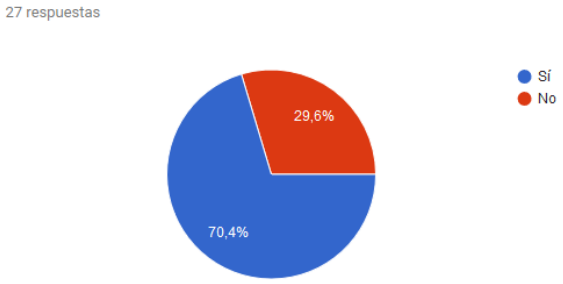
1. ¿Se ha presentado problemas relacionados al proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?




Resultado: El encuestado recibió previamente información en su correo acerca del proceso de gestión de pedidos, por ende, de las 27 personas encuestadas el 55,6% indican que si existe problemas relacionados al proceso de gestión de pedidos y el 44,4% indican que no existe inconvenientes.

**Pregunta 2:**

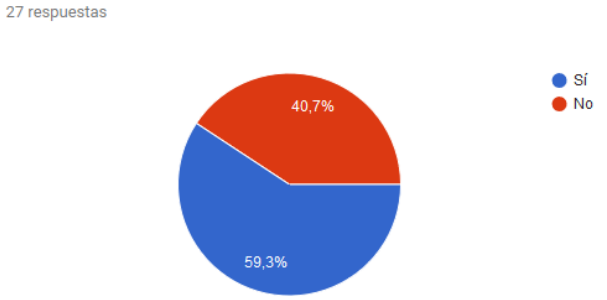
2. ¿Ha existido inconvenientes en la escasez de menús o cartas físicas cuando el restaurante se encuentra lleno en su totalidad, dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?




Resultado: De las 27 personas encuestadas el 70,4% indican que si existe inconvenientes en la escasez de menús o cartas físicas cuando el restaurante se encuentra lleno en su totalidad y el 29,6% indican que no existe inconvenientes.

**Pregunta 3:**

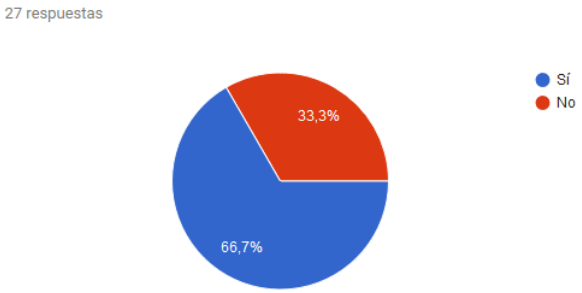
3. ¿Ha existido inconvenientes en la toma de un pedido a los clientes dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?



Resultado: De las 27 personas encuestadas el 59,3% indican que si existe inconvenientes en la toma de un pedido a los clientes y el 40,7% indican que no existe inconvenientes.

**Pregunta 4:**

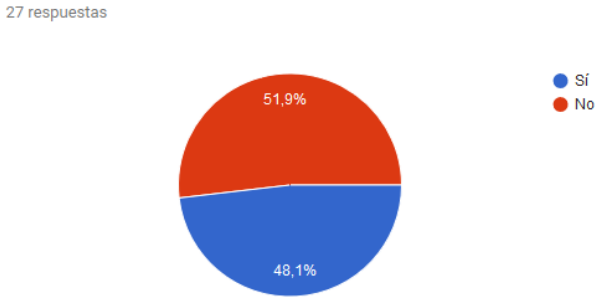
4. ¿Se ha presentado inconsistencias y retrasos en la entrega del pedido dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?




Resultado: De las 27 personas encuestadas el 66,7% indican que si se presenta inconsistencias y retrasos en la entrega de un pedido y el 33,3% indican que no existe problemas.

**Pregunta 5:**

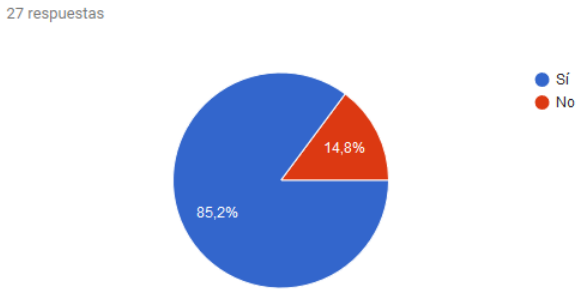
5. ¿Ha existido problemas en la publicación del menú dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?




Resultado: De las 27 personas encuestadas el 48,1% indican que no ha existido problemas en la publicación del menú dentro del proceso de gestión de pedidos y el 51,9% indican que si existe problemas.

**Pregunta 6:**

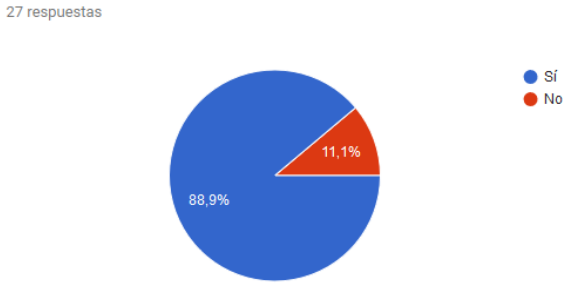
6. ¿Cree que sería necesario aplicar herramientas tecnológicas en el proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?




Resultado: El encuestado recibió previamente información en su correo acerca de las herramientas tecnológicas, por ende, de las 27 personas encuestadas el 85,2% indican que si sería necesario aplicar herramientas tecnológicas en el proceso de gestión de pedidos y el 14,8% indican que no es necesario de herramientas tecnológicas.

**Pregunta 7:**

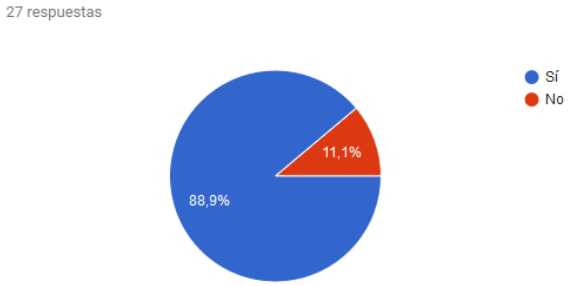
7. ¿Cree que sería necesaria la implementación de una aplicación Web y que a su vez sea Progresiva para fortalecer el proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?




Resultado: El encuestado recibió previamente información en su correo acerca de las Aplicaciones Web Progresivas, por ende, de las 27 personas encuestadas el 88,9% indican que si sería necesaria la implementación de una aplicación Web y que a su vez sea Progresiva y el 11,1% indican que no es necesario la implementación de una aplicación Web.

**Pregunta 8:**

8. ¿Cree que sería ventajoso que la aplicación Web Progresiva funcione dentro y fuera del establecimiento con o sin conexión a internet dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?

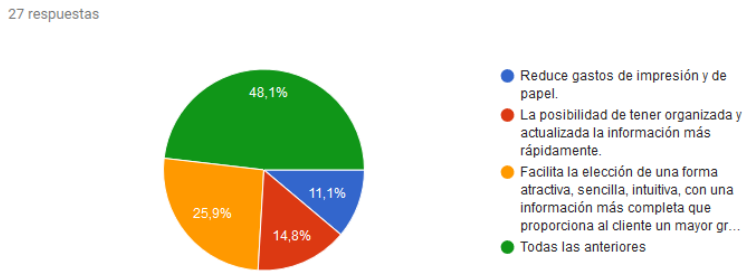



Resultado: El encuestado recibió previamente información en su correo acerca del funcionamiento de una Aplicación Web Progresiva y de su funcionamiento en dos estados online y offline, por ende, de las 27 personas encuestadas el 88,9% indican que si sería ventajoso que la aplicación Web Progresiva funcione dentro y fuera del establecimiento con o sin conexión a internet y el 11,1% indican que no sería ventajoso.

**Pregunta 9:**

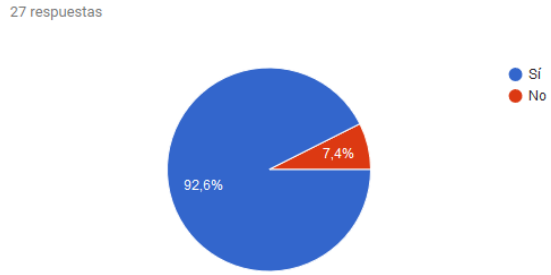
9. ¿Indique los beneficios más relevantes que tendría la implementación de una aplicación Web Progresiva dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?

- Reduce gastos de impresión y de papel.
- La posibilidad de tener organizada y actualizada la información más rápidamente.
- Facilita la elección de una forma atractiva, sencilla, intuitiva, con una información más completa que proporciona al cliente un mayor grado de acierto y satisfacción con su selección.




**Pregunta 10:**

10. ¿Estaría dispuesto a utilizar una aplicación Web Progresiva que permita optimizar el proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?




## 3. Aplicación del método Chi cuadrado en la investigación

El método de chi-cuadrada es una prueba de hipótesis que compara la distribución observada de los datos con una distribución esperada de los datos (Minitab, 2019) de manera estadística, comprobable con un fundamento de investigación basado en la formulación de una hipótesis.

Es decir que la teoría del chi cuadrado dispone determinar si dos variables están relacionadas o no, por ende, es necesario escribir una hipótesis nula y una hipótesis alternativa.

### 3.1. Formulación de hipótesis en la investigación

**Nula($H_0$):** El uso de una arquitectura para la construcción de una Aplicación Web Progresiva es independiente y no fortalece el proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant en la Ciudad de Otavalo.

**Alternativa($H_1$):** El uso de una arquitectura para la construcción de una Aplicación Web Progresiva no es independiente y fortalece el proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant en la Ciudad de Otavalo.

### 3.2. Tabla de Frecuencias esperadas

Para calcular todos y cada uno de los valores de la tabla de frecuencia se debe realizar las sumas por filas y columnas y la suma total como se muestra en la Tabla 26.

| Pregunta | SI | NO | |
|---|---|---|---|
| Pregunta 1 | 12 | 15 | 27 |
| Pregunta 2 | 19 | 8 | 27 |
| Pregunta 3 | 16 | 11 | 27 |
| Pregunta 4 | 18 | 9 | 27 |
| Pregunta 5 | 13 | 14 | 27 |
| Pregunta 6 | 23 | 4 | 27 |
| Pregunta 7 | 24 | 3 | 27 |
| Pregunta 8 | 24 | 3 | 27 |
| Pregunta 10 | 25 | 2 | 27 |
| | 174 | 69 | 243 |

**Tabla 26:** Tabla de frecuencias observadas

**Fuente:** Propia

Posterior al cálculo de frecuencias se realizó la aplicación de la ecuación para determinar los valores de las frecuencias esperadas, es decir multiplicar el total de las columnas por el total de las filas y ese resultado dividirlo para la suma total como se muestra en la Tabla 27.

$$= \frac{total\ columna\ x\ total\ fila}{suma\ total}$$

| | |
|---|---|
| 19,33 | 7,67 |
| 19,33 | 7,67 |
| 19,33 | 7,67 |
| 19,33 | 7,67 |
| 19,33 | 7,67 |
| 19,33 | 7,67 |
| 19,33 | 7,67 |
| 19,33 | 7,67 |
| 19,33 | 7,67 |

**Tabla 27***:* Tabla de frecuencias esperadas

**Fuente:** Propia

### 3.3. Aplicando Chi Cuadrado

Para obtener el valor de chi cuadrado se calculó utilizando, el valor de la frecuencia observada menos el valor de la frecuencia esperada todo ese valor elevado al cuadrado y posteriormente dividiéndolo para la frecuencia del valor esperado, el cálculo se lo realiza para cada valor de la tabla de frecuencias observadas como se detalla en la siguiente fórmula:

$$\chi^2_{calc} = \sum \frac{(f_0 - f_e)^2}{f_e}$$

$f_0$ : *Frecuencia del valor observado.*

$f_e$ : *Frecuencia del valor esperado.*

$$X^2calc = \frac{(12-19.33)^2}{19.33} + \frac{(15-7.67)^2}{7.67} + \frac{(19-19.33)^2}{19.33} + \frac{(8-7.67)^2}{7.67} + \frac{(16-19.33)^2}{19.33} + \frac{(11-7.67)^2}{7.67} + \frac{(18-19.33)^2}{19.33} + \frac{(9-7.67)^2}{7.67} + \frac{(13-19.33)^2}{19.33} + \frac{(14-7.67)^2}{7.67} + \frac{(23-19.33)^2}{19.33} + \frac{(4-7.67)^2}{7.67} + \frac{(24-19.33)^2}{19.33} + \frac{(3-7.67)^2}{7.67} + \frac{(24-19.33)^2}{19.33} + \frac{(3-7.67)^2}{7.67} + \frac{(25-19.33)^2}{19.33} + \frac{(2-7.67)^2}{7.67}$$

$$X^2calc = 2.78 + 7.01 + 0.01 + 0.01 + 0.57 + 1.45 + 0.09 + 0.23 + 2.07 + 5.22 + 0.70 + 1.76 + 1.13 + 2.84 + 1.13 + 2.84 + 1.66 + 4.19$$

$$\boldsymbol{X^2calc = 35.69}$$

Por un lado, para el cálculo del grado de libertad se multiplica la cantidad de filas menos uno por la cantidad de columnas menos uno como se muestra en la siguiente ecuación:

$v = (Cantidad\ de\ filas - 1)(Cantidad\ de\ columnas - 1)$

$v = (9-1)(2-1)$

$v = 8$

Por otro lado, el nivel de significancia es el error para poder rechazar la hipótesis nula siendo verdadera, por lo general el grado de significancia es de 0.05 que muestra que existe una probabilidad del 0.95 de que la hipótesis nula sea verdadera. Por ende, para la investigación se aplicó trabajar con un nivel de significancia de 0.05.

Finalmente se verifica en la tabla de valores de Chi Cuadrado Crítico de acuerdo al nivel de significancia(p=0.05) y el grado de libertad(v=8) detallada en el siguiente esquema de valores como se muestra en la Tabla 28:

P = Probabilidad de encontrar un valor mayor o igual que el chi cuadrado tabulado, v = Grados de Libertad

| v/p | 0,001 | 0,0025 | 0,005 | 0,01 | 0,025 | 0,05 | 0,1 | 0,15 | 0,2 | 0,25 | 0,3 | 0,35 | 0,4 | 0,45 | 0,5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10,8274 | 9,1404 | 7,8794 | 6,6349 | 5,0239 | 3,8415 | 2,7055 | 2,0722 | 1,6424 | 1,3233 | 1,0742 | 0,8735 | 0,7083 | 0,5707 | 0,4549 |
| 2 | 13,8150 | 11,9827 | 10,5965 | 9,2104 | 7,3778 | 5,9915 | 4,6052 | 3,7942 | 3,2189 | 2,7726 | 2,4079 | 2,0996 | 1,8326 | 1,5970 | 1,3863 |
| 3 | 16,2660 | 14,3202 | 12,8381 | 11,3449 | 9,3484 | 7,8147 | 6,2514 | 5,3170 | 4,6416 | 4,1083 | 3,6649 | 3,2831 | 2,9462 | 2,6430 | 2,3660 |
| 4 | 18,4662 | 16,4238 | 14,8602 | 13,2767 | 11,1433 | 9,4877 | 7,7794 | 6,7449 | 5,9886 | 5,3853 | 4,8784 | 4,4377 | 4,0446 | 3,6871 | 3,3567 |
| 5 | 20,5147 | 18,3854 | 16,7496 | 15,0863 | 12,8325 | 11,0705 | 9,2363 | 8,1152 | 7,2893 | 6,6257 | 6,0644 | 5,5731 | 5,1319 | 4,7278 | 4,3515 |
| 6 | 22,4575 | 20,2491 | 18,5475 | 16,8119 | 14,4494 | 12,5916 | 10,6446 | 9,4461 | 8,5581 | 7,8408 | 7,2311 | 6,6948 | 6,2108 | 5,7652 | 5,3481 |
| 7 | 24,3213 | 22,0402 | 20,2777 | 18,4753 | 16,0128 | 14,0671 | 12,0170 | 10,7479 | 9,8032 | 9,0371 | 8,3834 | 7,8061 | 7,2832 | 6,8000 | 6,3458 |
| 8 | 26,1239 | 23,7742 | 21,9549 | 20,0902 | 17,5345 | 15,5073 | 13,3616 | 12,0271 | 11,0301 | 10,2189 | 9,5245 | 8,9094 | 8,3505 | 7,8325 | 7,3441 |
| 9 | 27,8767 | 25,4625 | 23,5893 | 21,6660 | 19,0228 | 16,9190 | 14,6837 | 13,2880 | 12,2421 | 11,3887 | 10,6564 | 10,0060 | 9,4136 | 8,8632 | 8,3428 |
| 10 | 29,5879 | 27,1119 | 25,1881 | 23,2093 | 20,4832 | 18,3070 | 15,9872 | 14,5339 | 13,4420 | 12,5489 | 11,7807 | 11,0971 | 10,4732 | 9,8922 | 9,3418 |
| 11 | 31,2635 | 28,7291 | 26,7569 | 24,7250 | 21,9200 | 19,6752 | 17,2750 | 15,7671 | 14,6314 | 13,7007 | 12,8987 | 12,1836 | 11,5298 | 10,9199 | 10,3410 |
| 12 | 32,9092 | 30,3182 | 28,2997 | 26,2170 | 23,3367 | 21,0261 | 18,5493 | 16,9893 | 15,8120 | 14,8454 | 14,0111 | 13,2661 | 12,5838 | 11,9463 | 11,3403 |
| 13 | 34,5274 | 31,8830 | 29,8193 | 27,6882 | 24,7356 | 22,3620 | 19,8119 | 18,2020 | 16,9848 | 15,9839 | 15,1187 | 14,3451 | 13,6356 | 12,9717 | 12,3398 |
| 14 | 36,1239 | 33,4262 | 31,3194 | 29,1412 | 26,1189 | 23,6848 | 21,0641 | 19,4062 | 18,1508 | 17,1169 | 16,2221 | 15,4209 | 14,6853 | 13,9961 | 13,3393 |
| 15 | 37,6978 | 34,9494 | 32,8015 | 30,5780 | 27,4884 | 24,9958 | 22,3071 | 20,6030 | 19,3107 | 18,2451 | 17,3217 | 16,4940 | 15,7332 | 15,0197 | 14,3389 |
| 16 | 39,2518 | 36,4555 | 34,2671 | 31,9999 | 28,8453 | 26,2962 | 23,5418 | 21,7931 | 20,4651 | 19,3689 | 18,4179 | 17,5646 | 16,7795 | 16,0425 | 15,3385 |
| 17 | 40,7911 | 37,9462 | 35,7184 | 33,4087 | 30,1910 | 27,5871 | 24,7690 | 22,9770 | 21,6146 | 20,4887 | 19,5110 | 18,6330 | 17,8244 | 17,0646 | 16,3382 |
| 18 | 42,3119 | 39,4220 | 37,1564 | 34,8052 | 31,5264 | 28,8693 | 25,9894 | 24,1555 | 22,7595 | 21,6049 | 20,6014 | 19,6993 | 18,8679 | 18,0860 | 17,3379 |
| 19 | 43,8194 | 40,8847 | 38,5821 | 36,1908 | 32,8523 | 30,1435 | 27,2036 | 25,3289 | 23,9004 | 22,7178 | 21,6891 | 20,7638 | 19,9102 | 19,1069 | 18,3376 |
| 20 | 45,3142 | 42,3358 | 39,9969 | 37,5663 | 34,1696 | 31,4104 | 28,4120 | 26,4976 | 25,0375 | 23,8277 | 22,7745 | 21,8265 | 20,9514 | 20,1272 | 19,3374 |
| 21 | 46,7963 | 43,7749 | 41,4009 | 38,9322 | 35,4789 | 32,6706 | 29,6151 | 27,6620 | 26,1711 | 24,9348 | 23,8578 | 22,8876 | 21,9915 | 21,1470 | 20,3372 |
| 22 | 48,2676 | 45,2041 | 42,7957 | 40,2894 | 36,7807 | 33,9245 | 30,8133 | 28,8224 | 27,3015 | 26,0393 | 24,9390 | 23,9473 | 23,0307 | 22,1663 | 21,3370 |
| 23 | 49,7276 | 46,6231 | 44,1814 | 41,6383 | 38,0756 | 35,1725 | 32,0069 | 29,9792 | 28,4288 | 27,1413 | 26,0184 | 25,0055 | 24,0689 | 23,1852 | 22,3369 |
| 24 | 51,1790 | 48,0336 | 45,5584 | 42,9798 | 39,3641 | 36,4150 | 33,1962 | 31,1325 | 29,5533 | 28,2412 | 27,0960 | 26,0625 | 25,1064 | 24,2037 | 23,3367 |
| 25 | 52,6187 | 49,4351 | 46,9280 | 44,3140 | 40,6465 | 37,6525 | 34,3816 | 32,2825 | 30,6752 | 29,3388 | 28,1719 | 27,1183 | 26,1430 | 25,2218 | 24,3366 |
| 26 | 54,0511 | 50,8291 | 48,2898 | 45,6416 | 41,9231 | 38,8851 | 35,5632 | 33,4295 | 31,7946 | 30,4346 | 29,2463 | 28,1730 | 27,1789 | 26,2395 | 25,3365 |
| 27 | 55,4751 | 52,2152 | 49,6450 | 46,9628 | 43,1945 | 40,1133 | 36,7412 | 34,5736 | 32,9117 | 31,5284 | 30,3193 | 29,2266 | 28,2141 | 27,2569 | 26,3363 |
| 28 | 56,8918 | 53,5939 | 50,9936 | 48,2782 | 44,4608 | 41,3372 | 37,9159 | 35,7150 | 34,0266 | 32,6205 | 31,3909 | 30,2791 | 29,2486 | 28,2740 | 27,3362 |
| 29 | 58,3006 | 54,9662 | 52,3355 | 49,5878 | 45,7223 | 42,5569 | 39,0875 | 36,8538 | 35,1394 | 33,7109 | 32,4612 | 31,3308 | 30,2825 | 29,2908 | 28,3361 |

**Tabla 28:** Distribución de Chi Cuadrado $x^2$

**Fuente:** https://bit.ly/2WRnhq8

Si el valor de chi cuadrado calculado es menor o igual que el chi cuadrado crítico como consecuencia se acepta la hipótesis nula, caso contrario se acepta la hipótesis alternativa.

Para el caso de la investigación formulada se muestra la siguiente información:

$$X^2 \text{ calc} \geq \text{Valor crítico}$$

$$\boldsymbol{35.62 \geq 15.5073}$$

Como resultado de la aplicación del método se demuestra que el chi cuadrado calculado es mayor al chi cuadrado crítico, razón por la que se rechaza la hipótesis nula **$H_0$:** la cual es: El uso de una arquitectura para la construcción de una Aplicación Web Progresiva es independiente y no fortalece el proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant en la Ciudad de Otavalo, y se acepta la hipótesis alternativa **$H_1$:** El uso de una arquitectura para la construcción de una Aplicación Web Progresiva no es independiente y fortalece del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant en la Ciudad de Otavalo.

# CONCLUSIONES

- La escasez de métodos y fuentes bibliográficas para evaluar una arquitectura causo ciertos niveles de complejidad dentro de la investigación para escoger la arquitectura adecuada.
- Mediante el estudio de las tres arquitecturas para la construcción de Web Progresivas, se determinó que, App Shell es la arquitectura factible porque permite almacenar de forma local los datos de la aplicación a través de un service workers para ser utilizado cuando haya una baja señal de internet o inclusive sin ella.
- La aplicación de la Arquitectura permitió realizar una Web Progresiva, sin embargo, no es posible incluir una base de datos, porque al compilar la aplicación es necesario construirla implicando duplicar el proyecto, es decir que, para la base de datos resulta complicada hacer ese proceso porque la herramienta en la que se construyen una aplicación Web Progresiva trabaja con base de datos que se encuentra en la nube y que no funcionan sin conexión a internet.

  Por ende, no es eficiente gestionar la información de la base de datos dando como resultado que el prototipo implementado cumpla con la automatización del proceso de gestión de pedidos de tipo café restaurant a nivel informativo y no transaccional.

# RECOMENDACIONES

- Continuar con el estudio acerca de la arquitectura de una Web Progresiva con el objetivo de incluir un motor de base datos que permita desarrollar una aplicación transaccional.
- Incrementar fuentes bibliográficas en el campus Universitario referente a la carrera de Ingeniería de Software para permitir que la realización de futuras tesis con respecto al desarrollo de Web Progresivas sea más eficiente.
- Para la validación de los resultados se recomienda aplicar métodos formales
- Motivar el desarrollo de Web Progresivas dentro de los contenidos académicos para generar una transición natural hacia entornos móviles para que de esta manera los estudiantes tengan conocimiento de que existe un entorno entre las aplicaciones web nativas e híbridas como lo es las aplicaciones Web Progresivas

# REFERENCIAS


Alfsan. (25 de Mayo de 2012). *Arquitectura N capas*. Obtenido de Arquitectura N capas: http://iutll-abdd.blogspot.com/2012/05/arquitectura-de-n-capas.html

Androides, 3. (18 de Agosto de 2017). *Androides*. Obtenido de Androides: https://www.3androides.com/actualidad/100-que-es-el-desarrollo-de-aplicaciones-nativas

Bermeo, P. E. (2012). Análisis de la Arquitectura de desarrollo de sistemas N-Capas. Quito-Ecuador.

Carrillo, A. C. (2015). Diseño y Validación de Arquitecturas de Aplicaciones. *Revusta Ibérica de Sistemas y Tecnologías de la Información*, 13.

Cerpa, E. S. (2011). Arquitectura orientada a servicios para software de apoyo para el proceso personal de software. *Ingeniare.Revista chilena de ingenieros, vol19,* 40-52.

Cervantes, H., & Valencia, E. (2010). Diseño de la Arquitectura. *SG(Software Guru),* 15-20.

Chipantiza, V. L., & Olivo, B. E. (2015). *La usabilidad en el desarrollo de software.* Machala: UTMACH.

Contributors, E. (2013 de Julio de 2013). *Requisitos no funcionales*. Obtenido de Requisitos no funcionales: https://www.ecured.cu/index.php?title=Requisitos_no_funcionales&oldid=1995934

Ducot, R. (Mayo de 2017). *Survey Monkey*. Obtenido de https://es.surveymonkey.com/mp/likert-scale/

Fernández, S. (28 de Noviembre de 2016). *Xataka Móvil.* Obtenido de Xataka Móvil: https://www.xatakamovil.com/aplicaciones/el-debate-esta-servido-son-mejores-las-apps-con-version-web-o-las-apps-nativas

Gómez, V. (23 de Junio de 2017). *Instinto binario*. Obtenido de Instinto binario: https://instintobinario.com/arquitectura-en-tres-capas/

Hernández, U. (08 de Enero de 2017). *Código facilito.* Obtenido de Código facilito: https://codigofacilito.com/articulos/progressive-apps

IEEE, I. (2011). *Systems and software engineering —ISO/IEC/IEEE/42010.* Estados Unidos: Switzerland .

Jarquín, P. S. (2014). Ingeniería Web., (pág. 27). Nicaragua.

José Guillermo Valle, J. G. (2005). *Arquitectura Cliente-Servidor.* Colombia: Bello.

Maceda, H. C., & Careaga, P. V.-E. (2016). *ARQUITECTURA DE SOFTWARE.* México: Cengage Learning Editores, S.A.

Mejía, C. X. (2007). Tecnología SOA "Arquitectura Orientada a Servicios". Cuenca-Ecuador.

Molina, V. P. (2014). Manual de Procedimientos Operativos para Restaurantes de Comida Rápida. Cuenca-Ecuador.

Ortiz, E. N. (2017). Comparación de la Arquitectura N Capas Vs Arquitectura N Capas orientada al dominio con .Net, Aplicando en el sistema plan de fortalecimiento y mejoras de acreditación de la UNACH. Riobamba-Ecuador.

Partners, G. (2017). *Tecnologías Web avanzadas*. Obtenido de Tecnologías Web avanzadas: https://support.google.com/partners/answer/7336597?hl=es-419

Peña, F. (19 de Marzo de 2015). *ISO TOOLS-SOFTWARE DE GESTIÓN PARA LA EXCELENCIA EMPRESARIAL*. Obtenido de ISO TOOLS-SOFTWARE DE GESTIÓN PARA LA EXCELENCIA EMPRESARIAL: https://www.isotools.org/2015/03/19/que-son-las-normas-iso-y-cual-es-su-finalidad/

Pérez, M. (14 de octubre de 2016). *IEBS*. Obtenido de IEBS: https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarroladores-google-seo-sem/

Ralph, P., & Wand, Y. (2009). A Proposal for a Formal Definition of the Design Concept. En P. Ralph, & Y. Wand, *A Proposal for a Formal Definition of the Design Concept* (págs. 7-9). Canadá.

Rand, P. (15 de Octubre de 2016). *Hazhistoria.* Obtenido de Hazhistoria: https://www.hazhistoria.net/blog/%C2%BFqu%C3%A9-son-las-aplicaciones-web-progresivas-o-pwa

Rea, A. (2017). *"GUÍA METODOLÓGICA PARA LA GESTIÓN DE REQUERIMIENTOS DE DESARROLLO DE SOFTWARE DEL GOBIERNO AUTÓNOMO DESCENTRALIZADO.* Ibarra-Ecuador.

Robles, V. (15 de Enero de 2018). *Angular*. Obtenido de Angular: https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/

Rodríguez, F. (2018). *SOFT INGENIERÍA*. Obtenido de SOFT INGENIERÍA: http://soft-ingenieria.blogspot.com/p/blog-page_4991.html

Roger S. Pressman, P. (2010). *Ingeniería de Software-Un enfoque práctico.* México: Mc Graw Hi Educación.

Ruiz, M. (09 de Agosto de 2017). *OpenWebinars*. Obtenido de https://openwebinars.net/blog/que-es-firebase-de-google/

Santiago Domingo Moquillaza Henríquez, &. H. (2010). Programación en N capas. *Revista de Investigación de Sistemas e Informática*, 57-67.

Teja, G. M. (2003). *USABILIDAD Y ACCESIBILIDAD EN WEB.*

Valencia, U. T. (2016). *Introduccion a ATAM.* España.

Vera, M. (18 de Noviembre de 2013). *Inteligencia de Negocios*. Obtenido de Inteligencia de Negocios: http://www.i2btech.com/blog-i2b/tech-deployment/que-se-entiende-por-soa-y-cuales-son-sus-beneficios/

Vilanova, L. (2016). *Claves en la gestión de pedidos*. Obtenido de Claves en la gestión de pedidos: https://smarterworkspaces.kyocera.es/blog/claves-gestion-de-pedidos/

Vittone, J. C.-J. (2017). *Diseñando apps para móviles.* España-Barcelona: Maga&Seba.

Wiegers, K., & Beatty, J. (2013). *Software Requirements.* Estados Unidos: Best practices.

# ANEXOS

**Anexo1: Manual de funciones**

**a) Funciones del Administrador**

- Desarrolla actividades específicas de presupuesto para las áreas del establecimiento.
- Supervisa los presupuestos para controlar gastos.
- Se encarga de coordinar la publicidad del local.
- Se encarga de realizar y supervisar los horarios del personal.
- Trabaja en conjunto con el contador, en aspectos de pagos, impuestos, estados financieros.
- Delega y asigna diversas tareas.
- Revisa los reportes diarios de inventarios y compras.
- Revisa el uniforme del personal.
- Trabaja en todas las áreas del restaurante.

**b) Procedimientos del Servicio**

Todos estos elementos al trabajar en conjunto, consiguen el objetivo de tener una mejor eficiencia en la empresa cuando se lleve a cabo el proceso, y por ende la satisfacción del cliente (Molina, 2014).

Dentro de los procedimientos generales de servicios se toma en cuenta los siguientes aspectos:

- Limpieza:

  Dentro de este procedimiento se considera lo siguiente: baños limpios, uniformes limpios y aseo personal.
- Hospitalidad

  Dentro de este procedimiento se considera lo siguiente: venta sugerida; el mismo que permita ofrecer un producto más caro a alguna promoción, recibir al cliente con un saludo y sonrisa.
- Orden apropiada

  Hace referencia a dar al cliente lo que él desee, no insistir demasiado y si sobre todo la calidad del producto, es decir si un producto no está en perfectas condiciones para servirlo reemplazarlo por otro que si lo este.
- Mantenimiento

  Se refiere a tener mente preventiva con el fin de realizar chequeos periódicos a todos los equipos del establecimiento.
- Servicio rápido

Se refiere a cumplir con los tiempos ofrecidos en el caso de despachos en un rango de 15 minutos.

**c) Procedimientos del Área de servicio**

El personal encargado de esta área debe asegurarse de realizar tareas como (Molina, 2014):

- Ambientación del salón

  Tener en cuenta la limpieza y ambientar el restaurante con la iluminación apropiada, música, etc.

- Servicio de alimentos y bebidas

  Hace referencia a recibir a los clientes de la manera más amable con el fin de que se sientan cómodos y a gusto en el restaurante. También tomar las órdenes y entregarlas a tiempo para satisfacer al cliente.

- Operaciones después del servicio

  Se refiere a que el cliente cuando haya terminado de servirse, se lo despide invitándolo a regresar, recoger la vajilla al área de lavado y limpiar las mesas para el siguiente servicio.

**d) Procedimientos de limpieza del salón**

- Salón: la limpieza se la realizará de la siguiente manera:
  - Limpiar las mesas con un paño y desinfectante.
  - Limpiar los polvos de los cuadros u otros adornos, así como también los bordes de las ventanas.
  - Barrer el piso, asegurándose de limpiar debajo de las mesas y sillas.
  - Trapear el piso con desinfectante.

**e) Procedimientos generales de cocina**

Las personas que están encargadas de la cocina deben reunir cualidades como la que se menciona a continuación (Molina, 2014):

- Hábitos de seguridad y limpieza: al ser la función del cocinero preparar alimentos es indispensable la limpieza, esto comprende el aseo corporal, vestimenta (uniforme) apropiada, limpieza de herramientas, revisión, limpieza de recipientes y otros utensilios, despeje y limpieza del lugar de trabajo.

Funciones del cocinero

El cocinero debe cumplir con las siguientes funciones (Molina, 2014):

- Encargado de elaborar el menú.
- Hace las revisiones necesarias de la bodega

- Es el encargado de cuidar los equipos de cocina.
- Es el encargado de la limpieza de la cocina
- Es el encargado de llevar el inventario de cocina.
- Es el encargado de dar de baja los productos en mal estado.
- Encargado de preparar el menú.

Funciones del ayudante de cocina

El ayudante de cocina debe de cumplir con las siguientes funciones (Molina, 2014):

- Recibir la materia y guardarla.
- Encargado de despacho de las órdenes

**f) Procedimientos del Área de Caja**

En este caso al ser una microempresa la administradora es la encargada de la caja, por ende debe de tratar que el cliente se sienta a gusto desde su ingreso al restaurante, para esto debe recibir al cliente con Cortesía (buenos modales, amabilidad), Atención (personalizada), Respeto (usar correctamente el uniforme, ser paciente), y Entusiasmo (disfrutar del trabajo) (Molina, 2014).

Además, es el encargado de tomar las órdenes de los clientes, y de esta misma manera es quien trata de vender un mayor número de productos, para esto debe saber cómo sugerir otros artículos con eficiencia:

- Describir los ingredientes cuando se lo pregunten.
- No insistir si el cliente dice no.

Funciones del cajero:

- Conocer la carta.
- Aconsejar al cliente sobre los productos.
- Encargado de cobrar los pedidos del cliente
- Encargado de pasar las ordenes al área de cocina.
- Supervisar que los pedidos salgan correctamente y no demore mucho.
- Emitir las facturas a los clientes.

**g) Procedimientos para el área de compras**

El control de las existencias de materia prima es indispensable en un establecimiento de alimentos y bebidas, para ello es necesario establecer una serie de medidas que permitan en todo momento, tener un control total de las existencias que evitará pérdidas innecesarias y las compras se las realiza de acuerdo a lo que requiera el cocinero (Molina, 2014).

a) Funciones de Compras

En este caso al ser una microempresa la administradora es la encargada de las compras, debe de cumplir las siguientes funciones:

- Revisar el menú elaborado por el cocinero, para realizar las compras.
- Realizar las compras diarias.
- Realizar los pedidos a los proveedores.

## Anexo 2: Pruebas de iteraciones con las arquitecturas

### a. Responsive Design

- **Tercera iteración**

1. **Red:** Al ser hacer la tercera iteración se realiza en un tiempo de 116ms porque ya se encuentra en cache como se muestra en la Figura 52.

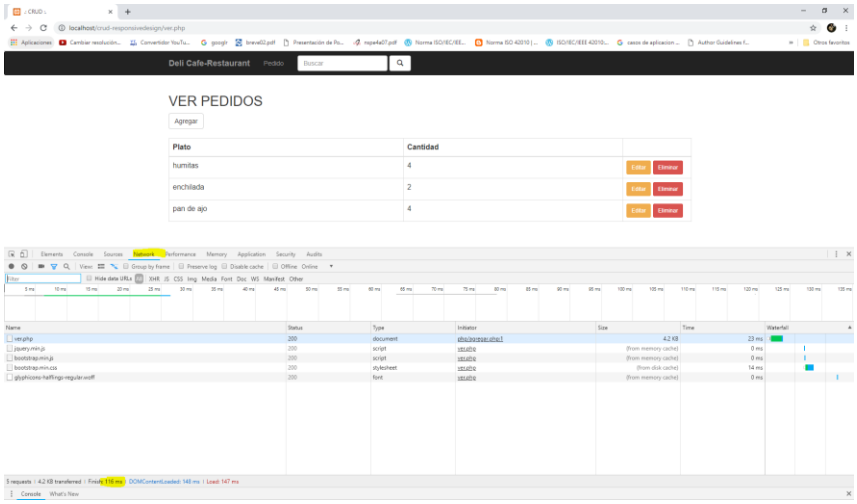



**Fig 52:** Arquitectura Responsive Design-Red

**Fuente:** Propia

2. **Memoria:** Al ser la segunda iteración consume 2595 kb; es decir que aumenta el consumo de memoria como se muestra en la Figura 32.

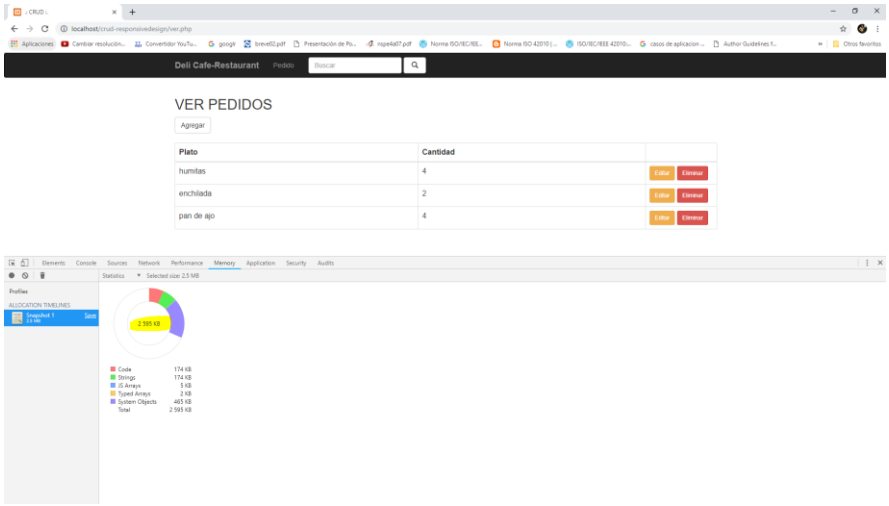



**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

3. **Rendimiento:** Al ser la segunda iteración se realiza en un tiempo 2415ms; es decir que aumenta el tiempo de rendimiento como se muestra en la Figura 33:

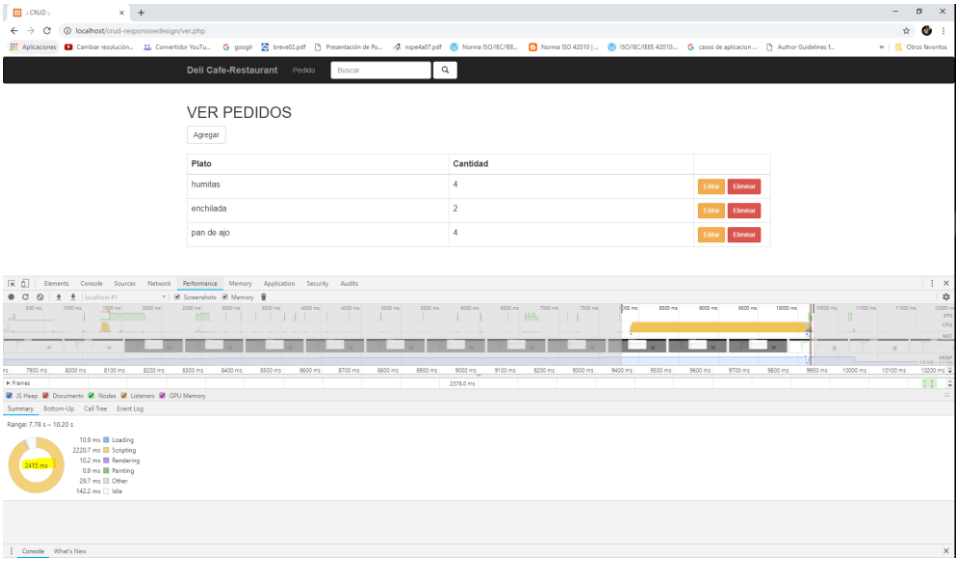



**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 4**

4. **Red:** Al ser hacer la tercera iteración se realiza en un tiempo de 117ms porque ya se encuentra en cache como se muestra en la Figura 31.

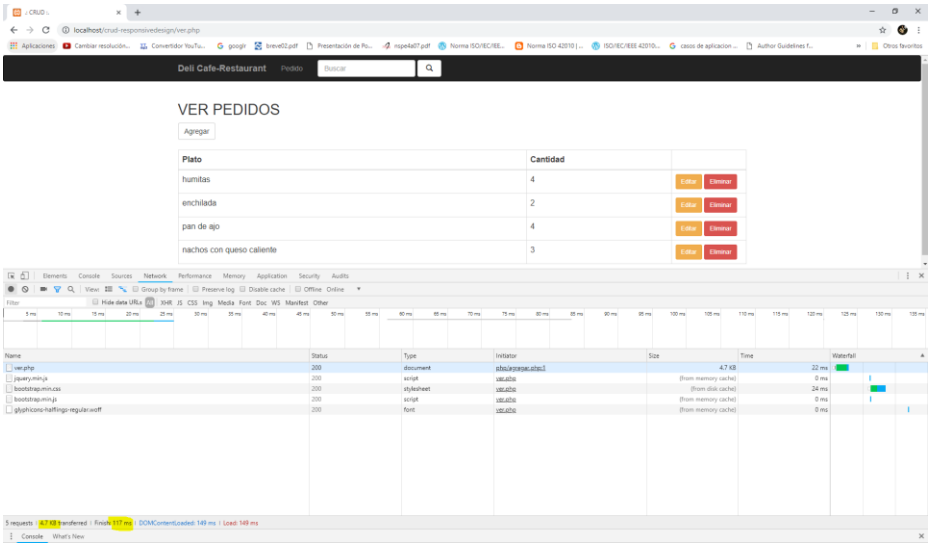



**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

5. **Memoria:** Al ser la segunda iteración consume 2487 kb; es decir que disminuye el consumo de memoria como se muestra en la Figura 32.

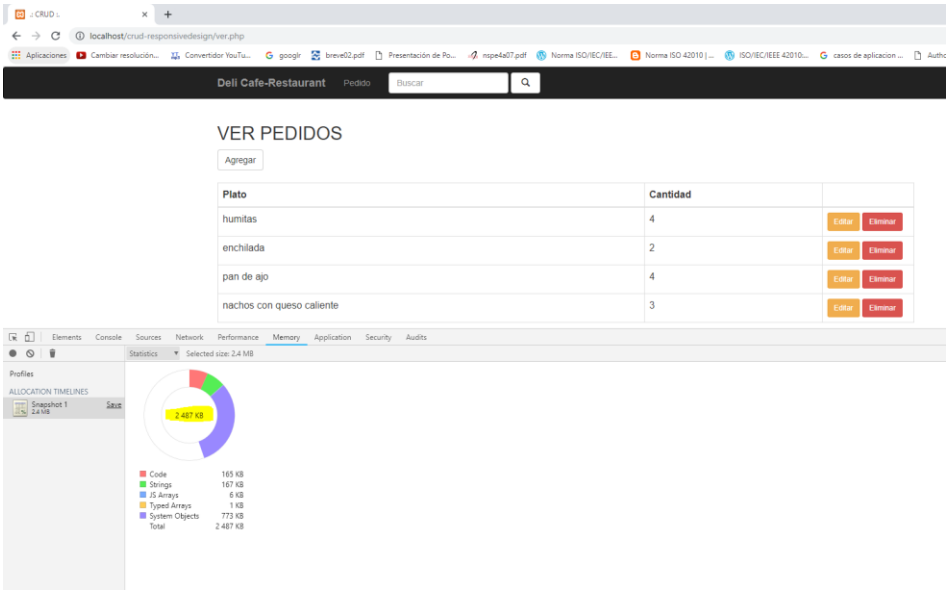



**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

6. **Rendimiento:** Al ser la segunda iteración se realiza en un tiempo 2259ms; es decir que baja el tiempo de rendimiento como se muestra en la Figura 33:

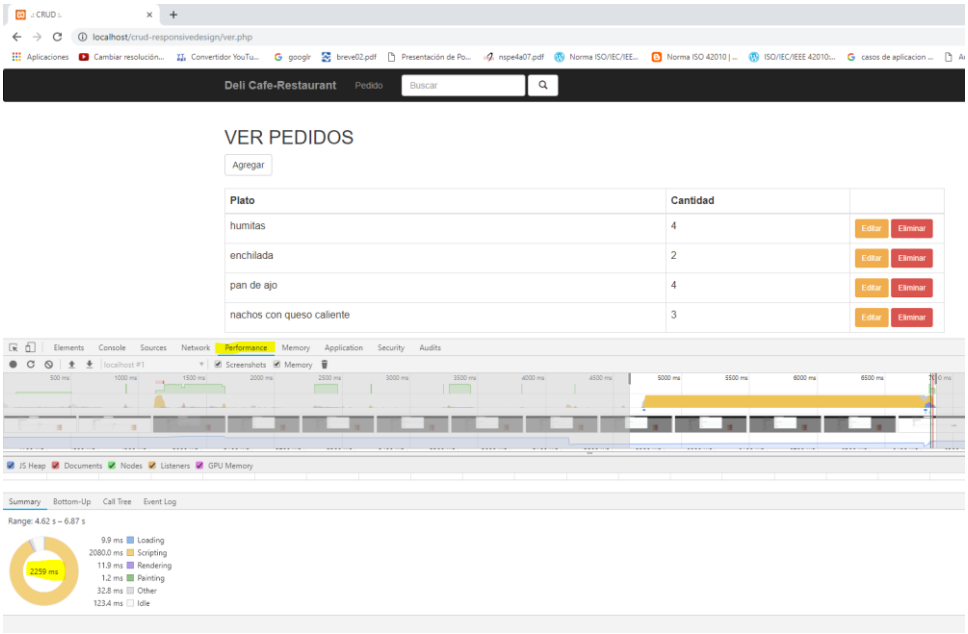



**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 5:**

7. **Red:** Al ser hacer la tercera iteración se realiza en un tiempo de 102ms porque ya se encuentra en cache como se muestra en la Figura 31.

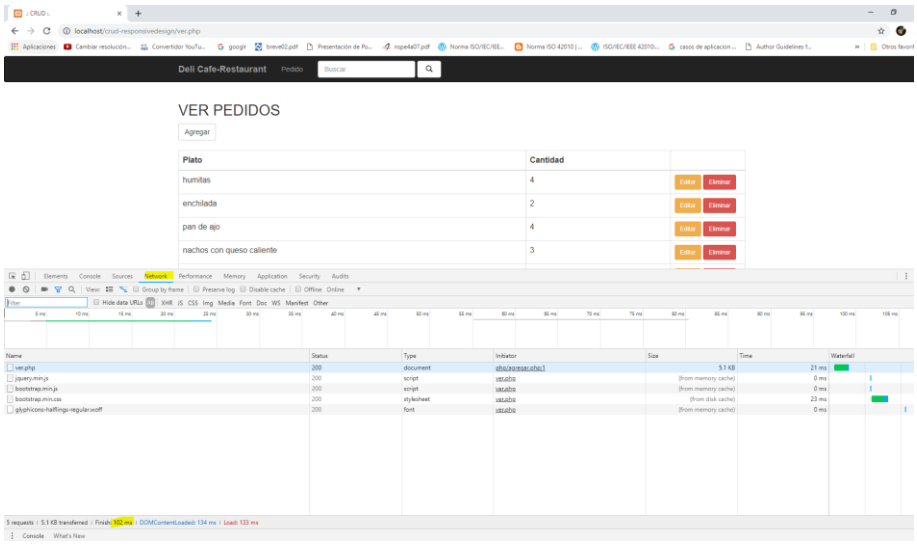



**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

8. **Memoria:** Al ser la segunda iteración consume 2498 kb; es decir que aumento el consumo de memoria como se muestra en la Figura 32.

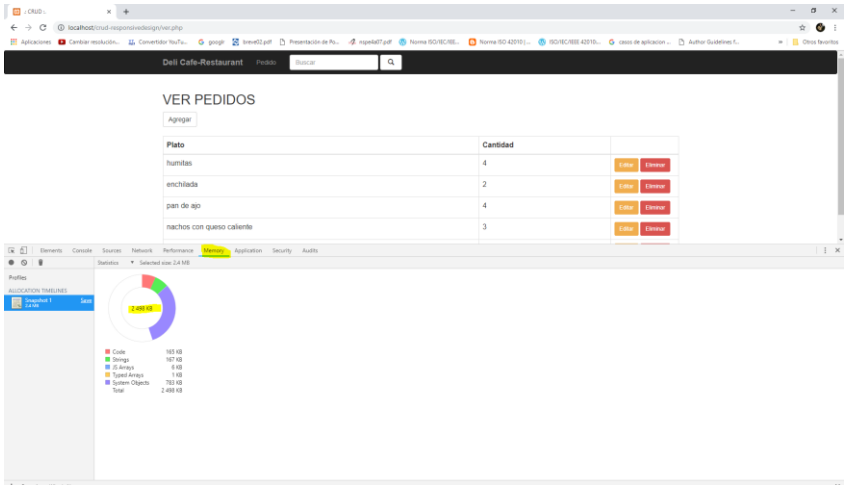



**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

9. **Rendimiento:** Al ser la segunda iteración se realiza en un tiempo 9353ms; es decir que baja el tiempo de rendimiento como se muestra en la Figura 33:

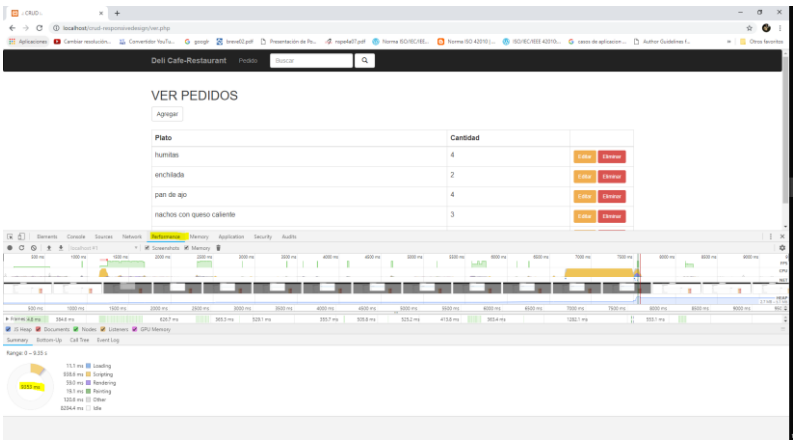



**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 6:**

10. **Red:** Al ser hacer la tercera iteración se realiza en un tiempo de 49ms porque ya se encuentra en cache como se muestra en la Figura 31.

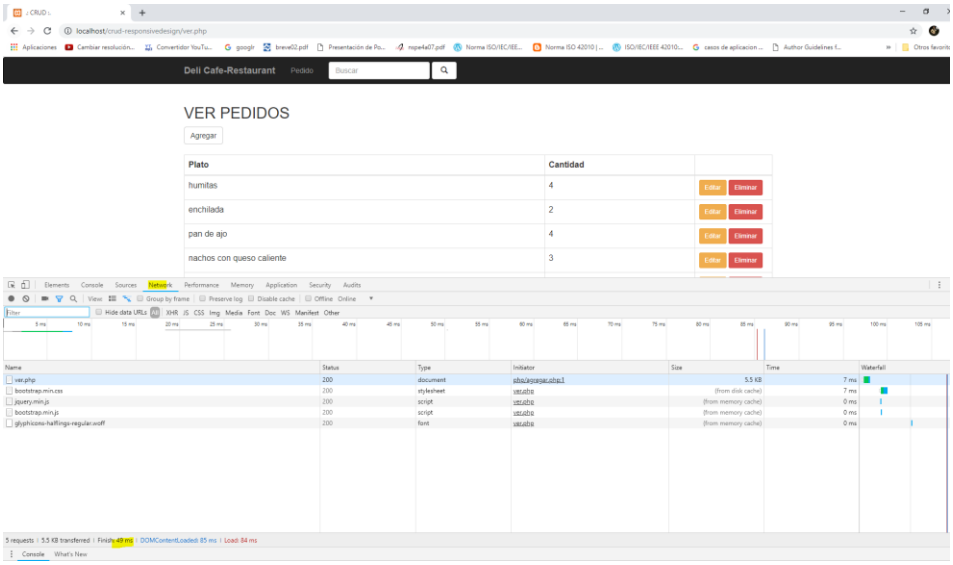



**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

11. **Memoria:** Al ser la segunda iteración consume 2293 kb; es decir que disminuye el consumo de memoria como se muestra en la Figura 32.

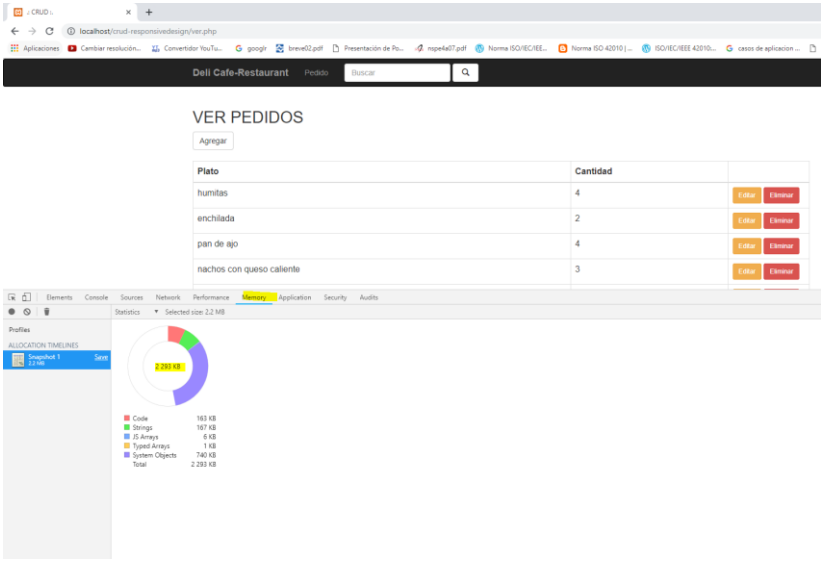



**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

12. **Rendimiento:** Al ser la segunda iteración se realiza en un tiempo 1932ms; es decir que baja el tiempo de rendimiento como se muestra en la Figura 33:

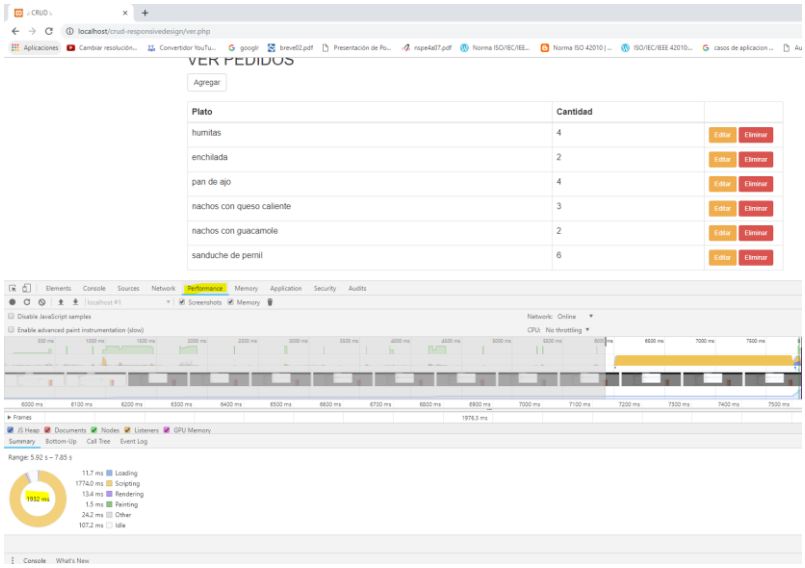



**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 7:**

13. **Red:** Al ser hacer la tercera iteración se realiza en un tiempo de 47ms porque ya se encuentra en cache como se muestra en la Figura 31.

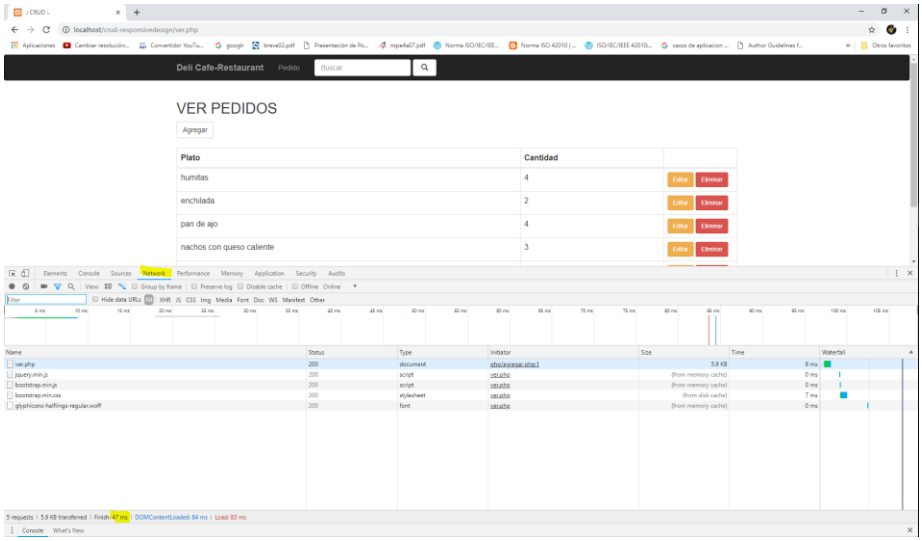



**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

14. **Memoria:** Al ser la segunda iteración consume 2685 kb; es decir que aumenta el consumo de memoria como se muestra en la Figura 32.

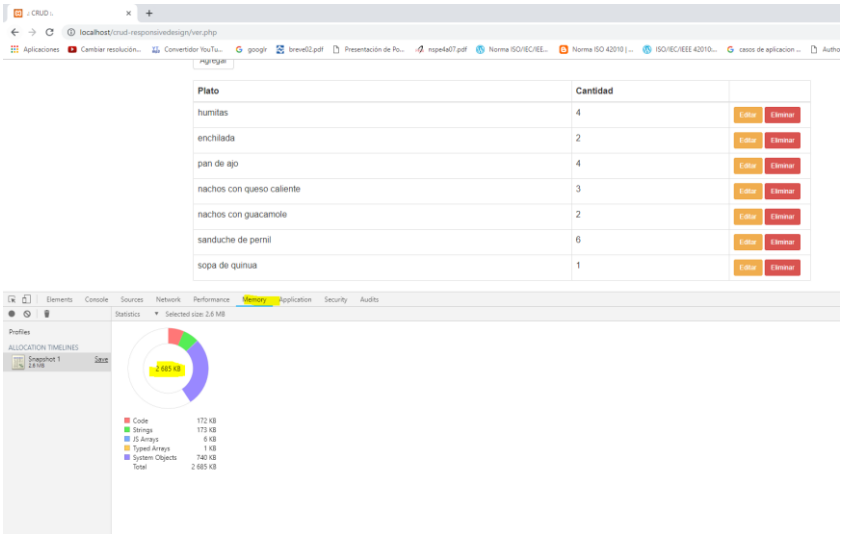



**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

15. **Rendimiento:** Al ser la segunda iteración se realiza en un tiempo 1643ms; es decir que baja el tiempo de rendimiento como se muestra en la Figura 33:

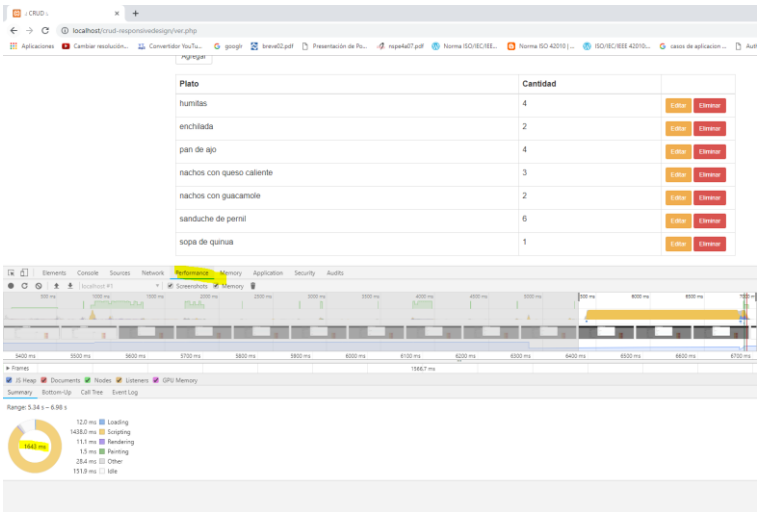



**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 8**

16. **Red:** Al ser hacer la tercera iteración se realiza en un tiempo de 43ms porque ya se encuentra en cache como se muestra en la Figura 31.

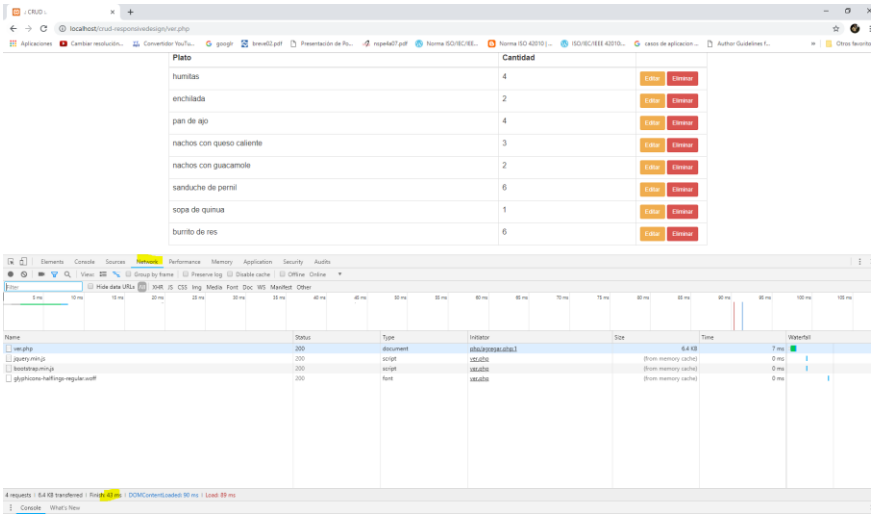



**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

17. **Memoria:** Al ser la segunda iteración consume 2335 kb; como se muestra en la Figura 32.

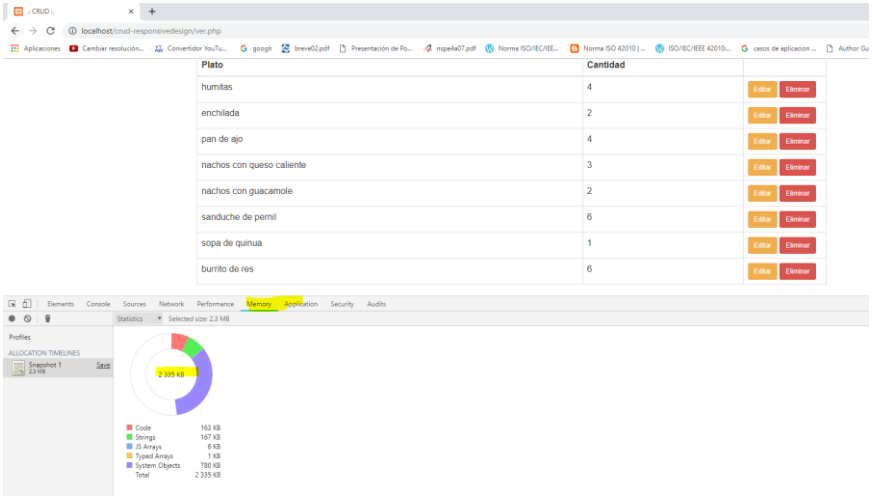



**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

18. **Rendimiento:** Al ser la segunda iteración se realiza en un tiempo 1448ms; como se muestra en la Figura 33:

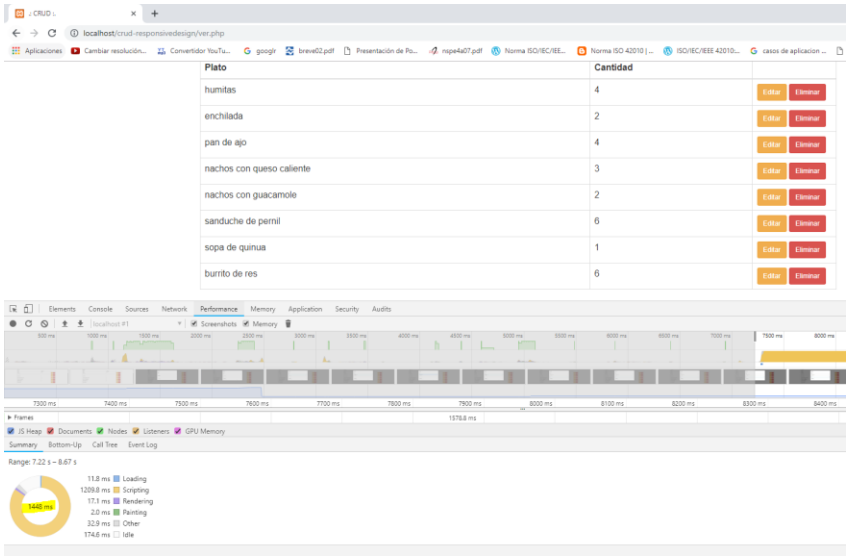



**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 9**

19. **Red:** Al hacer la siguiente iteración se realiza en un tiempo de 47ms porque ya se encuentra en cache como se muestra en la Figura 31.

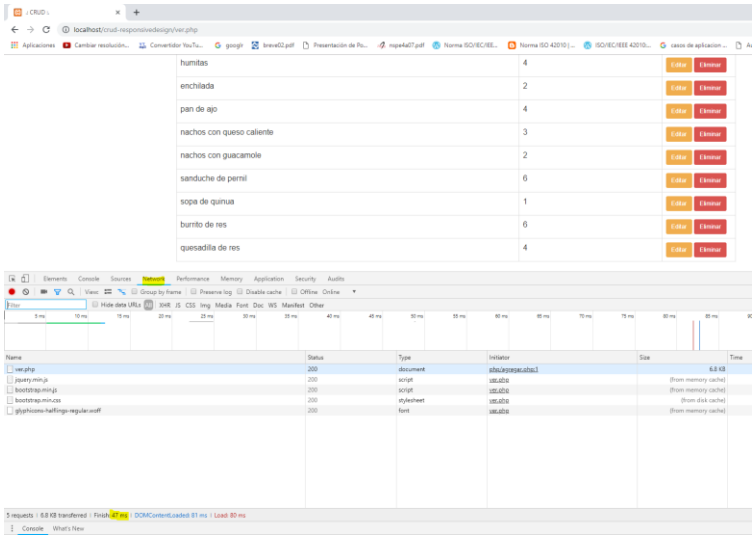



**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

20. **Memoria:** Al hacer la siguiente iteración consume 2336 kb; como se muestra en la Figura 32.

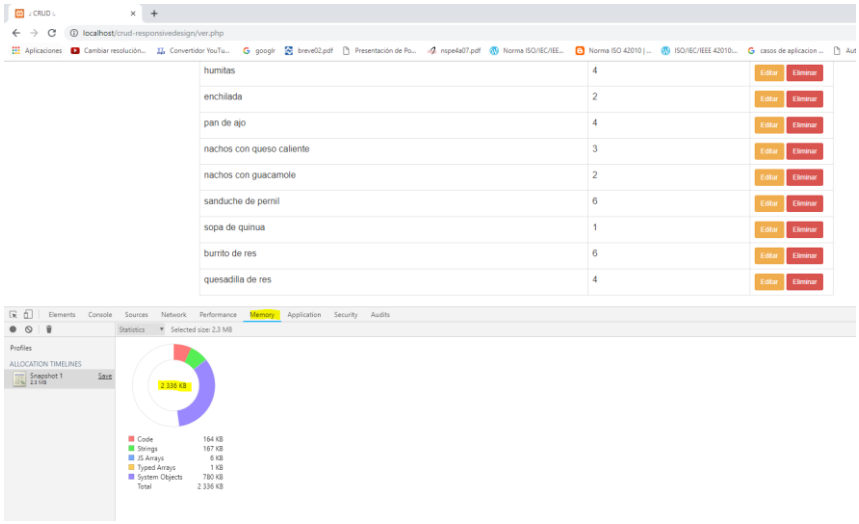



**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

21. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1213ms; como se muestra en la Figura 33:

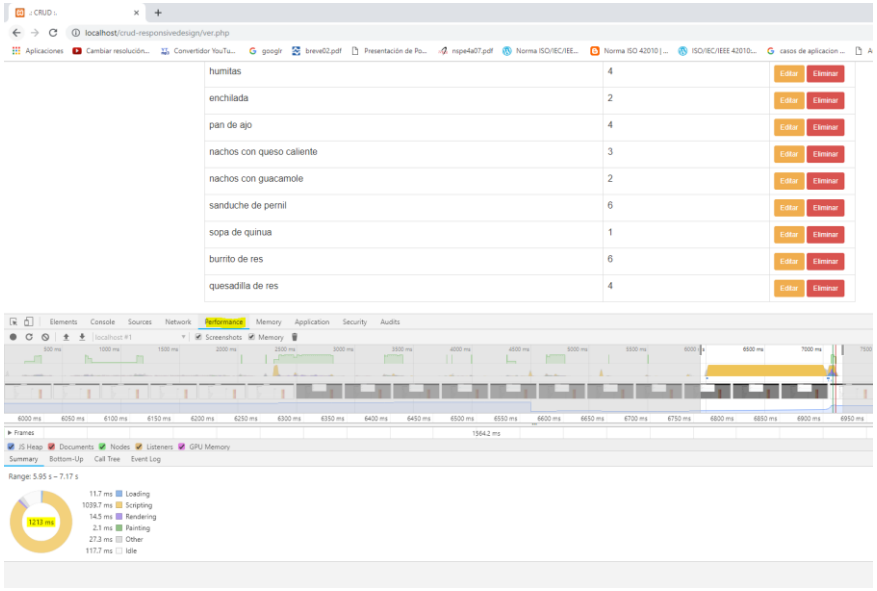



**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 10:**

22. **Red:** Al hacer la siguiente iteración se realiza en un tiempo de 71ms porque ya se encuentra en cache como se muestra en la Figura 31.

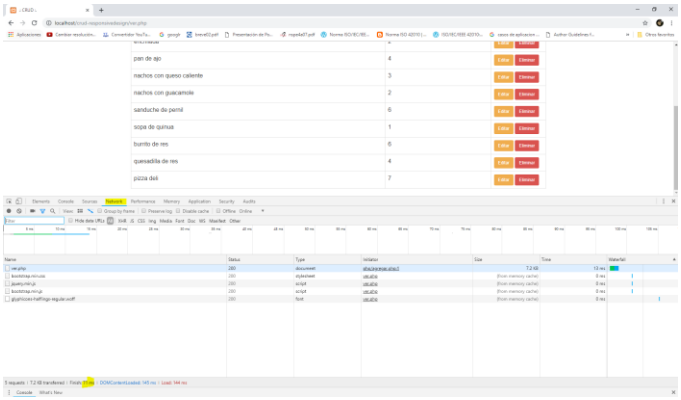


**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

**23. Memoria:** Al hacer la siguiente iteración consume 2714 kb; como se muestra en la Figura 32.

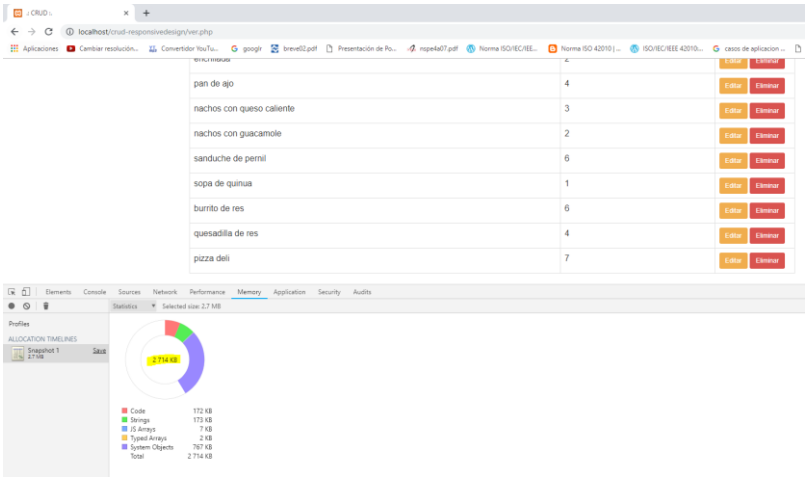



**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

**24. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1037 ms; como se muestra en la Figura 33:

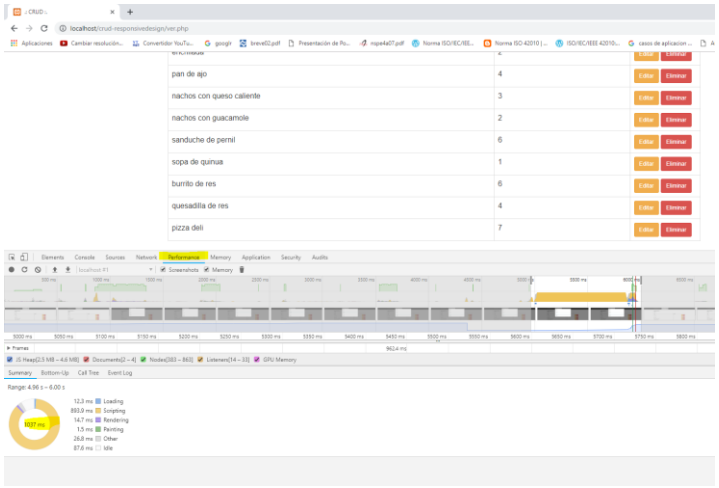



**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 11:**

**25. Red:** Al hacer la siguiente iteración se realiza en un tiempo de 50ms porque ya se encuentra en cache como se muestra en la Figura 31.

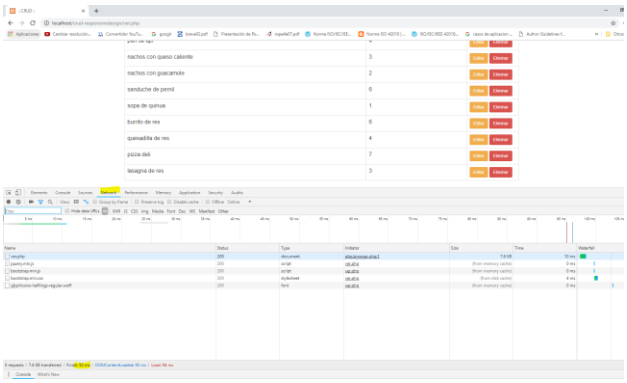


**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

**26. Memoria:** Al hacer la siguiente iteración consume 2717 kb como se muestra en la Figura 32.

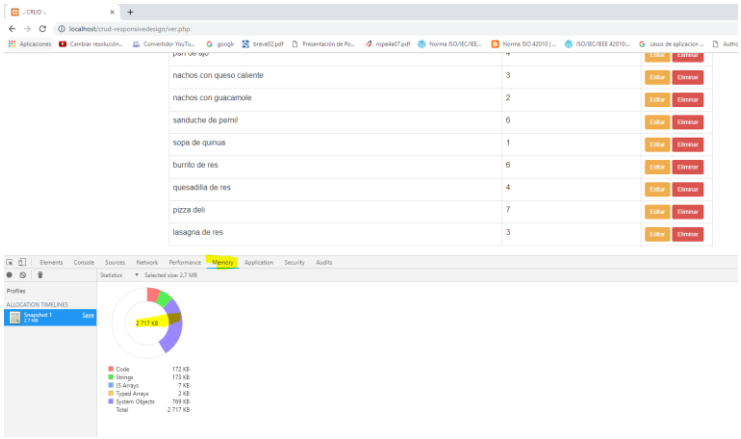


**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

27. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 2019 ms como se muestra en la Figura 33:

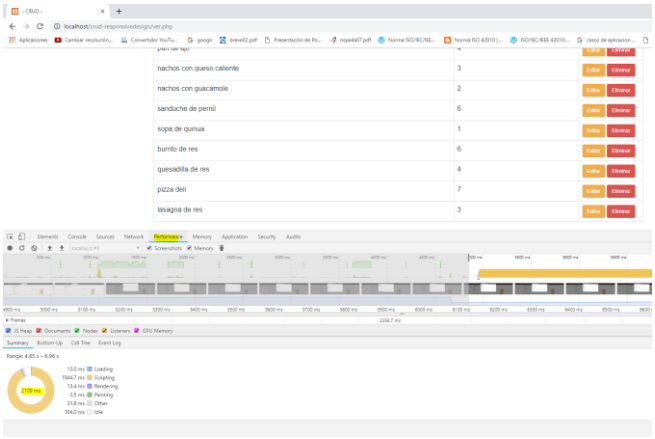


**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 12:**

28. **Red:** Al hacer la siguiente iteración se realiza en un tiempo de 78 ms porque ya se encuentra en cache como se muestra en la Figura 31.

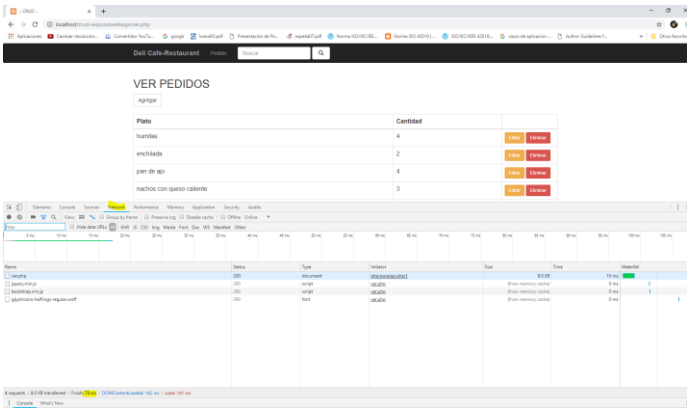


**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

29. **Memoria:** Al hacer la siguiente iteración consume 1252 kb como se muestra en la Figura 32.

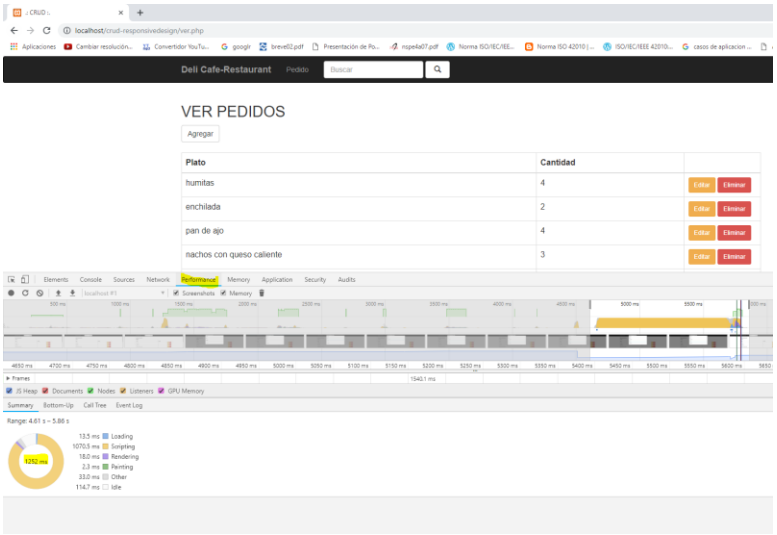



**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

30. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 2358 ms como se muestra en la Figura 33:

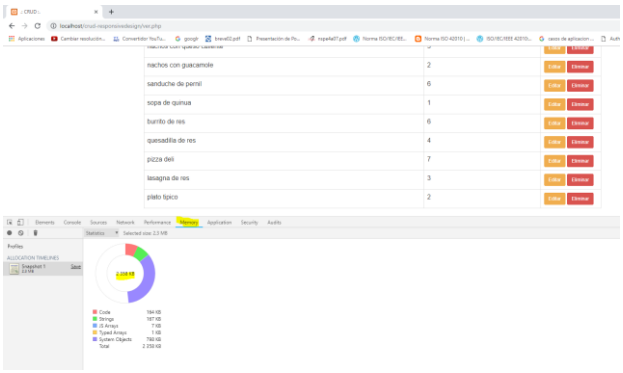


**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 13:**

31. **Red:** Al hacer la siguiente iteración se realiza en un tiempo de 45 ms porque ya se encuentra en cache como se muestra en la Figura 31.

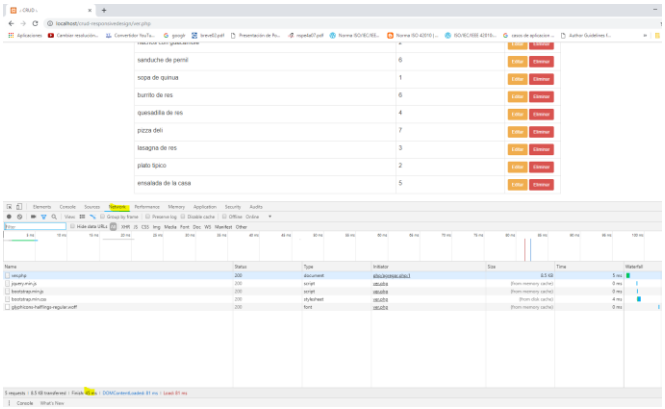


**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

32. **Memoria:** Al hacer la siguiente iteración consume 1252 kb como se muestra en la Figura 32.

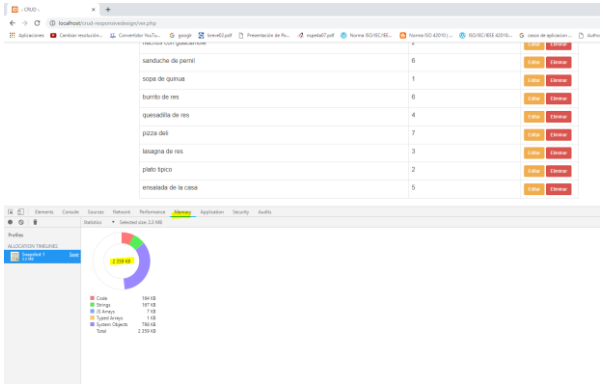


**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

33. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1702 ms como se muestra en la Figura 33:

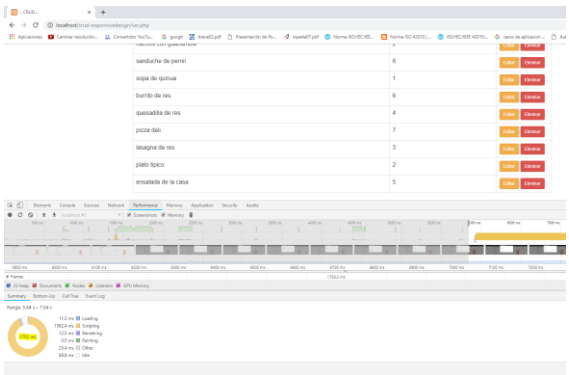


**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 14:**

34. **Red:** Al hacer la siguiente iteración se realiza en un tiempo de 70 ms porque ya se encuentra en cache como se muestra en la Figura 31.

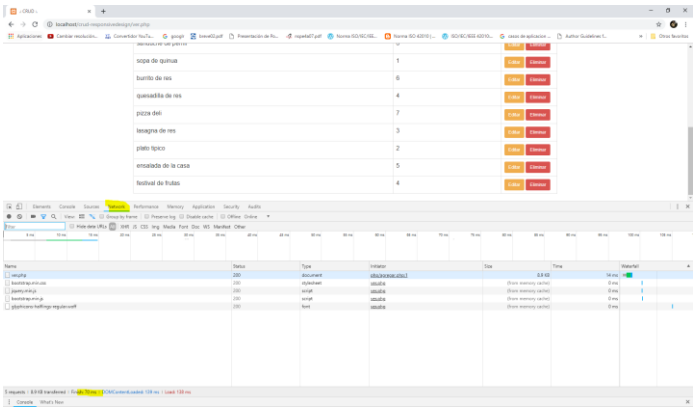


**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

35. **Memoria:** Al hacer la siguiente iteración consume 2382 kb como se muestra en la Figura 32.

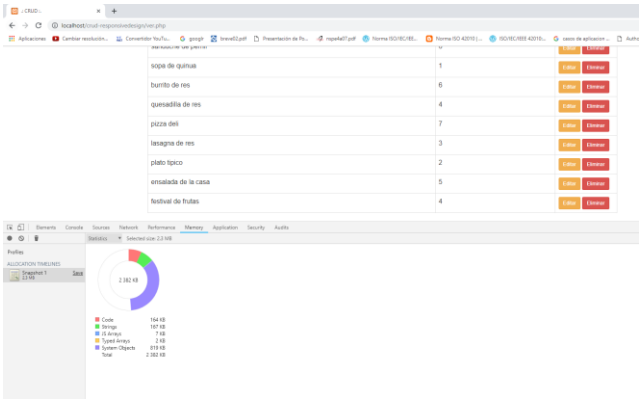


**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

36. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 2226ms como se muestra en la Figura 33:

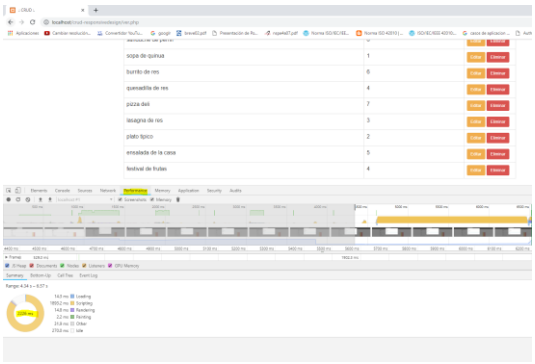


**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro.15 :**

37. **Red:** Al hacer la siguiente iteración se realiza en un tiempo de 76 ms porque ya se encuentra en cache como se muestra en la Figura 31.

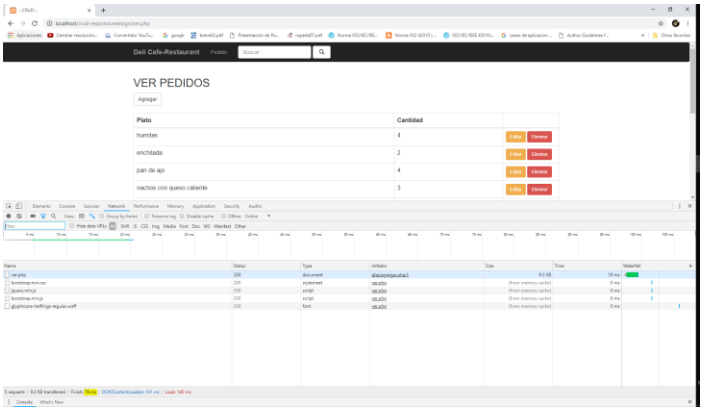



**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

38. **Memoria:** Al hacer la siguiente iteración consume 2383 kb como se muestra en la Figura 32.

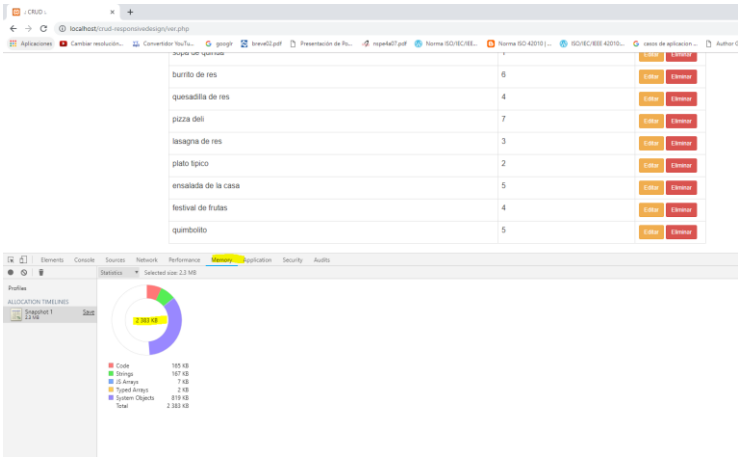



**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

39. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1106ms como se muestra en la Figura 33:

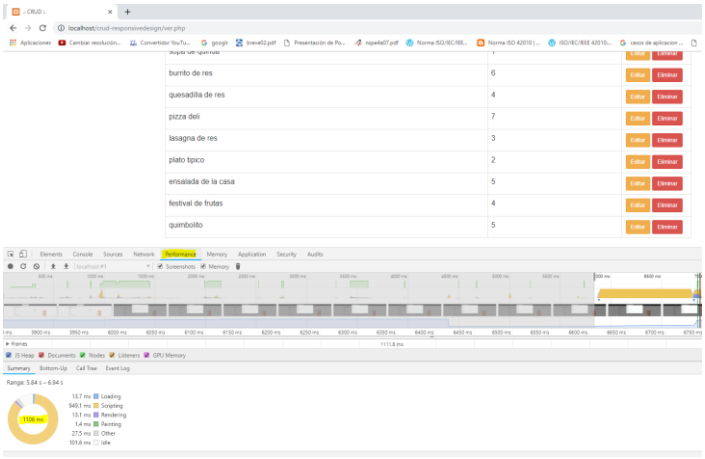



**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 16:**

**40. Red:** Al hacer la siguiente iteración se realiza en un tiempo de 71 ms porque ya se encuentra en cache como se muestra en la Figura 31.

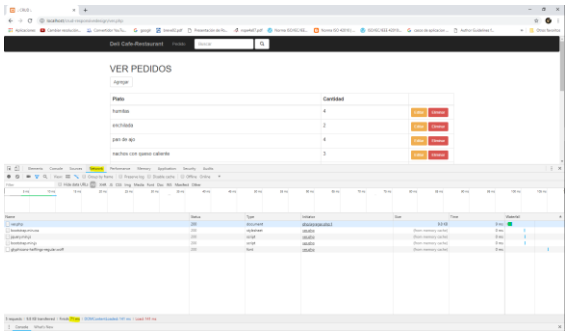



**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

**41. Memoria:** Al hacer la siguiente iteración consume 2387 kb como se muestra en la Figura 32.

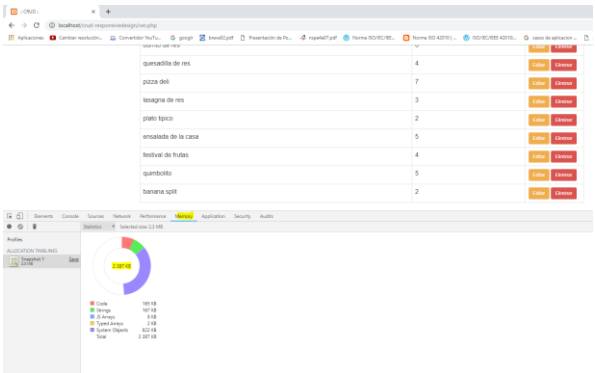



**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

**42. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1108ms como se muestra en la Figura 33:

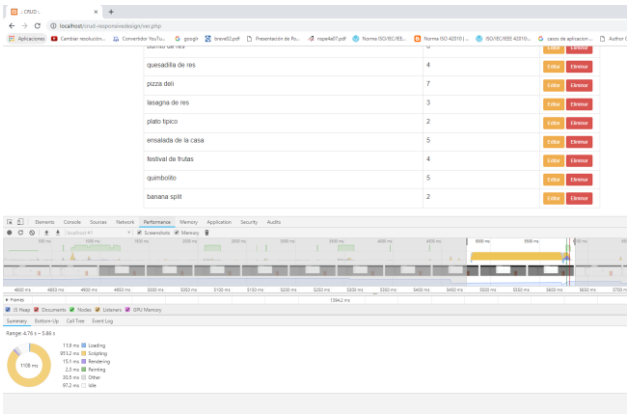


**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 17:**

**43. Red:** Al hacer la siguiente iteración se realiza en un tiempo de 64 ms porque ya se encuentra en cache como se muestra en la Figura 31.

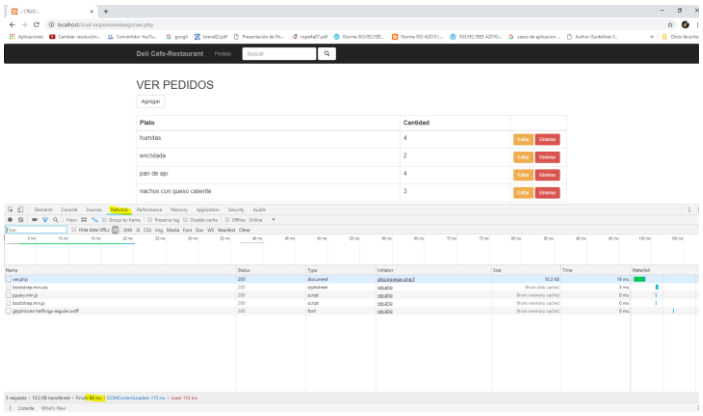


**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

**44. Memoria:** Al hacer la siguiente iteración consume 2389 kb como se muestra en la Figura 32.

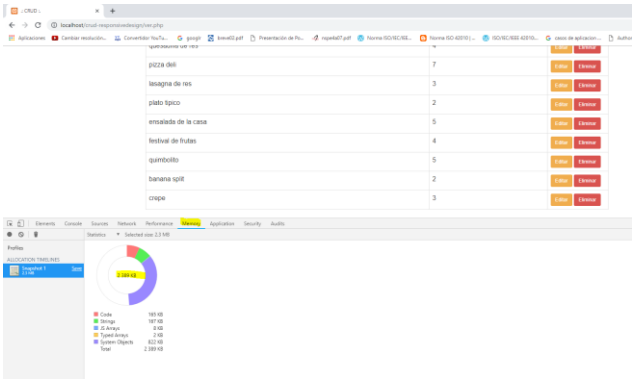


**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

45. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1645ms como se muestra en la Figura 33:

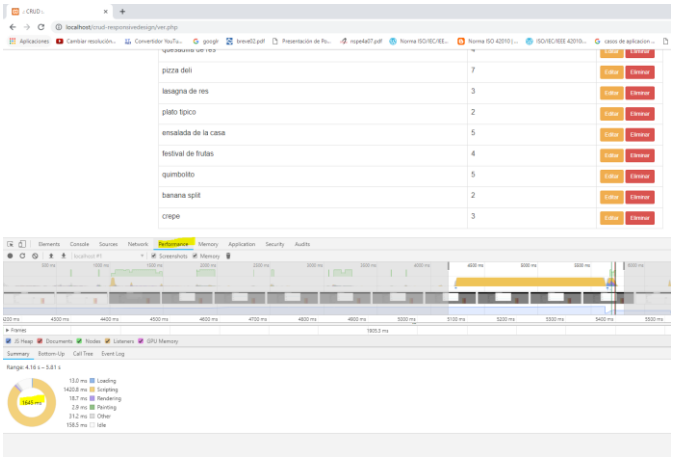


**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 18**

46. **Red:** Al hacer la siguiente iteración se realiza en un tiempo de 51 ms porque ya se encuentra en cache como se muestra en la Figura 31.

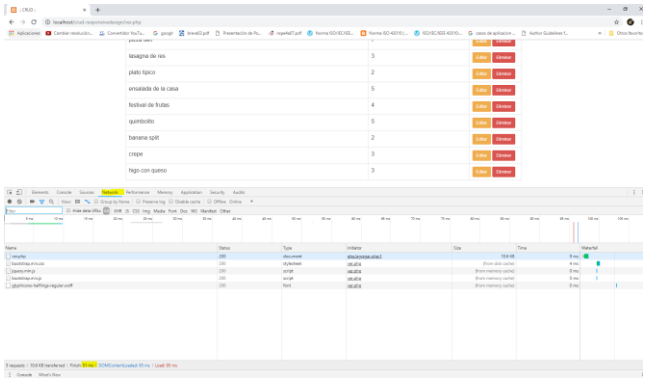


**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

47. **Memoria:** Al hacer la siguiente iteración consume 2390 kb como se muestra en la Figura 32.

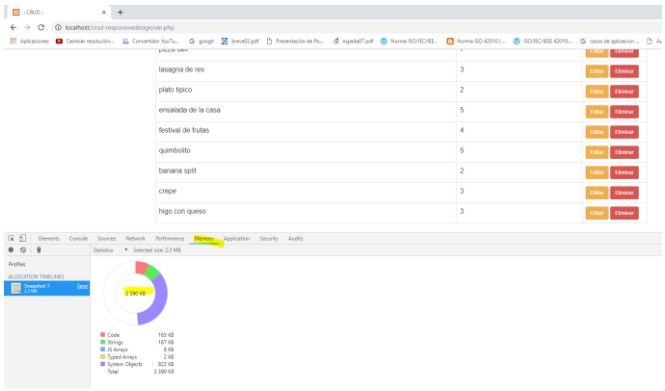


**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

48. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1146ms como se muestra en la Figura 33:

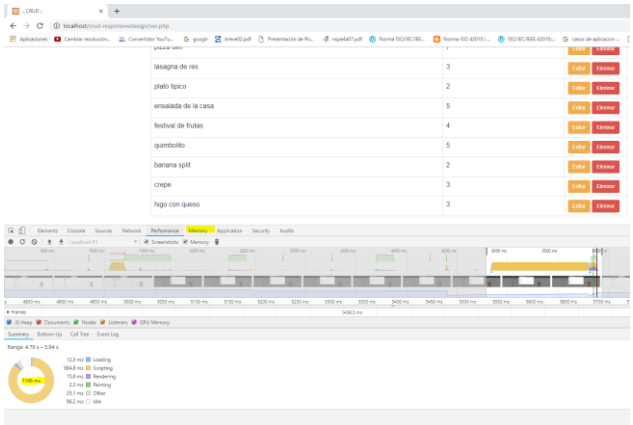


**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro. 19**

49. **Red:** Al hacer la siguiente iteración se realiza en un tiempo de 101 ms porque ya se encuentra en cache como se muestra en la Figura 31.

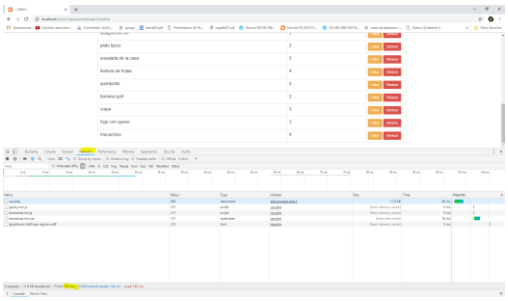


**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

50. **Memoria:** Al hacer la siguiente iteración consume 2423 kb como se muestra en la Figura 32.

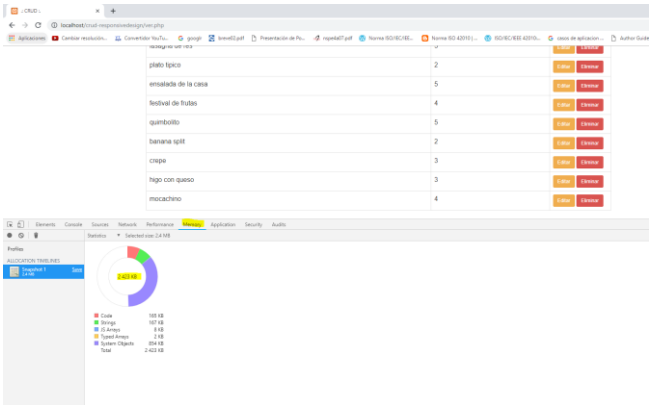


**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

51. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1192ms como se muestra en la Figura 33:

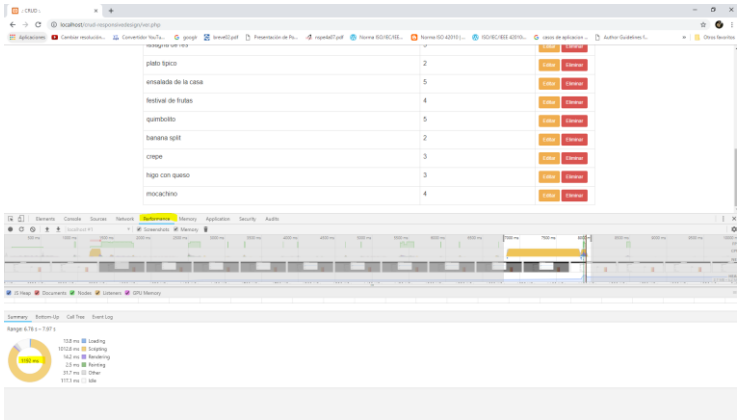


**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

- **Iteración nro.20:**

52. **Red:** Al hacer la siguiente iteración se realiza en un tiempo de 73 ms porque ya se encuentra en cache como se muestra en la Figura 31.

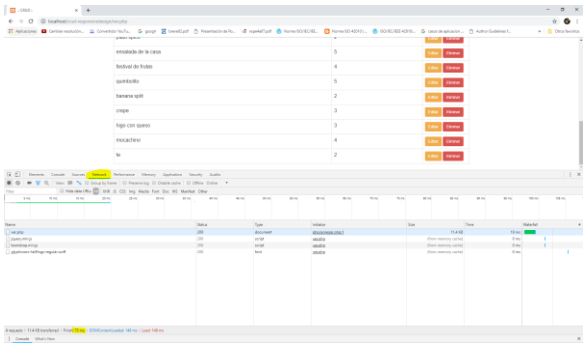


**Fig. 31.** Arquitectura Responsive Design-Red

**Fuente:** Propia

53. **Memoria:** Al hacer la siguiente iteración consume 2423 kb como se muestra en la Figura 32.

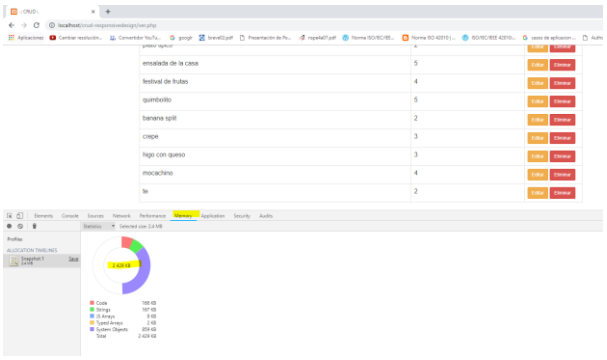


**Fig. 32.** Arquitectura Responsive Design-Memoria

**Fuente:** Propia

54. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1192ms como se muestra en la Figura 33:

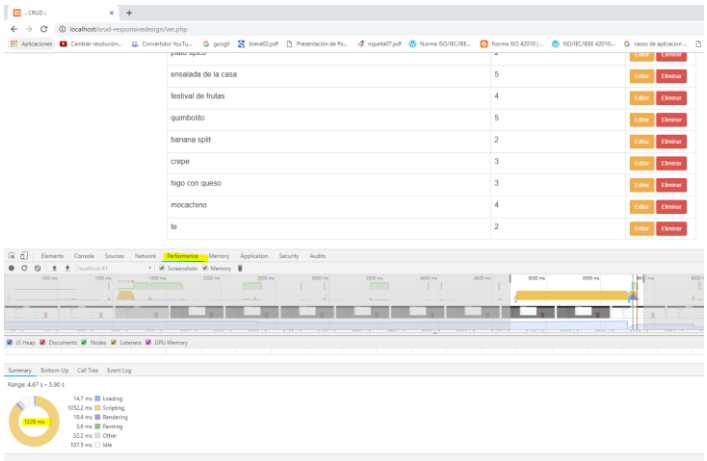


**Fig. 33.** Arquitectura Responsive Design-Rendimiento

**Fuente:** Propia

## b) SOA

- **Iteración nro. 3:**

**55. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 38ms como se muestra en la Figura 34:

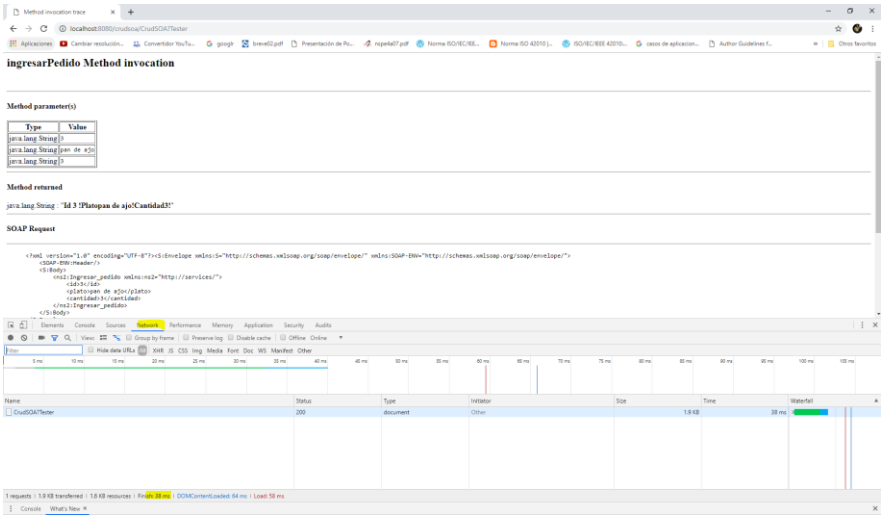



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**56. Memoria:** Al hacer la siguiente iteración consume 1853 kb como se muestra en la Figura 32.

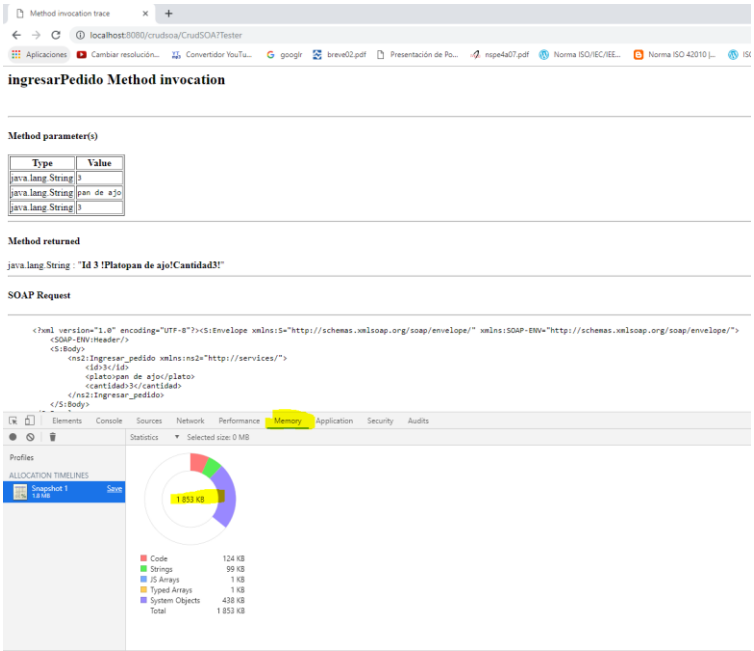


**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**57. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 3023ms como se muestra en la Figura 33:

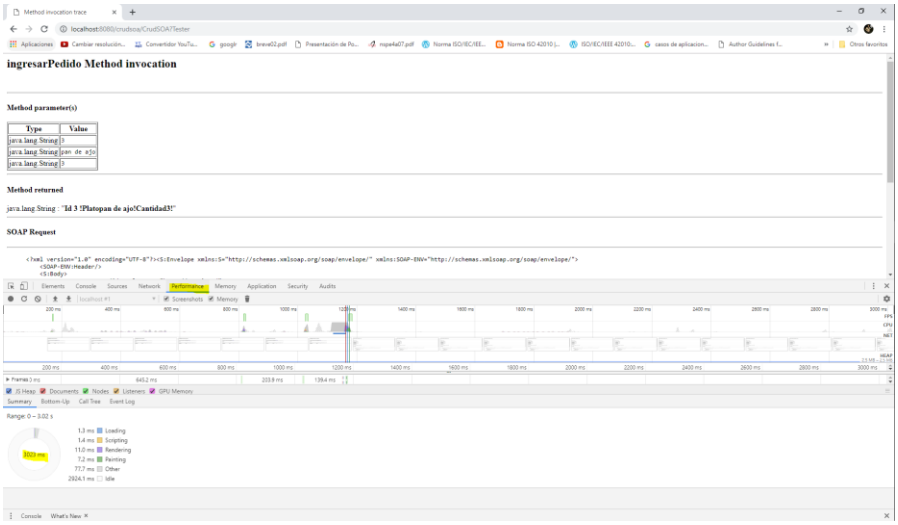



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 4:**

**58. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 23ms como se muestra en la Figura 34:

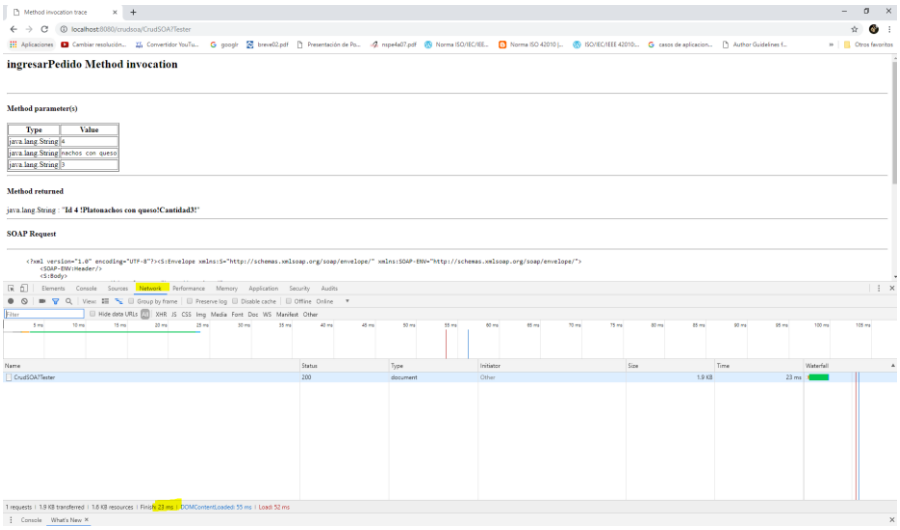



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**59. Memoria:** Al hacer la siguiente iteración consume 1830 kb como se muestra en la Figura 32.

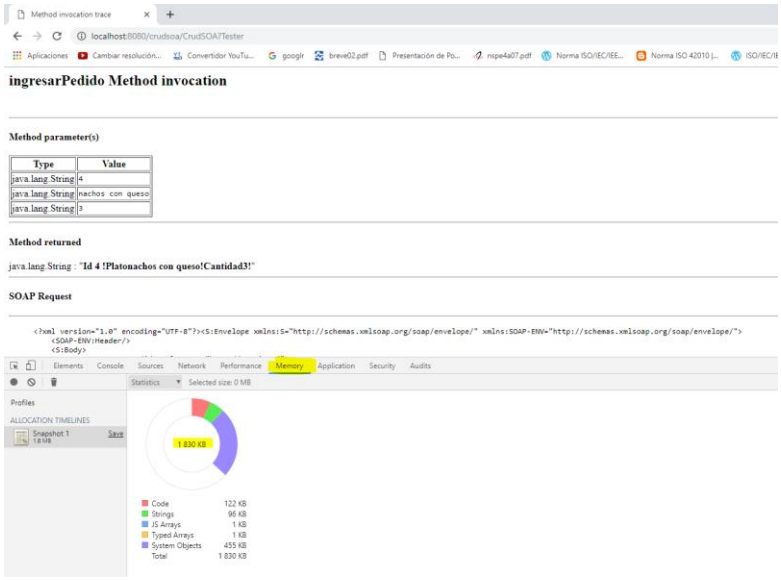



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

60. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 4373ms como se muestra en la Figura 33:

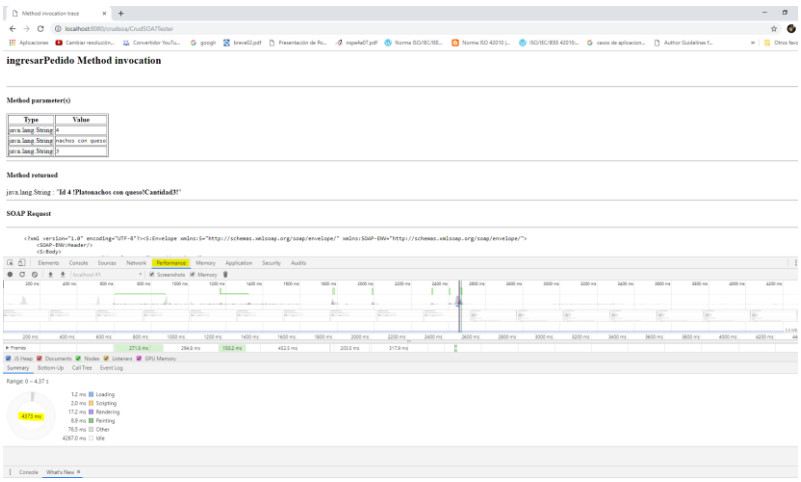



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 5:**

61. **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 25ms como se muestra en la Figura 34:

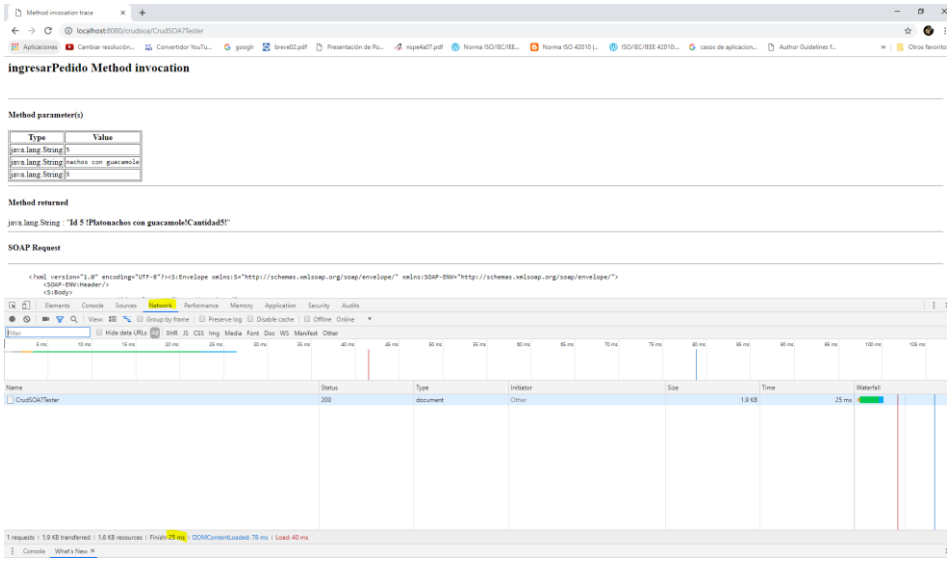



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**62. Memoria:** Al hacer la siguiente iteración consume 1307 kb como se muestra en la Figura 32.

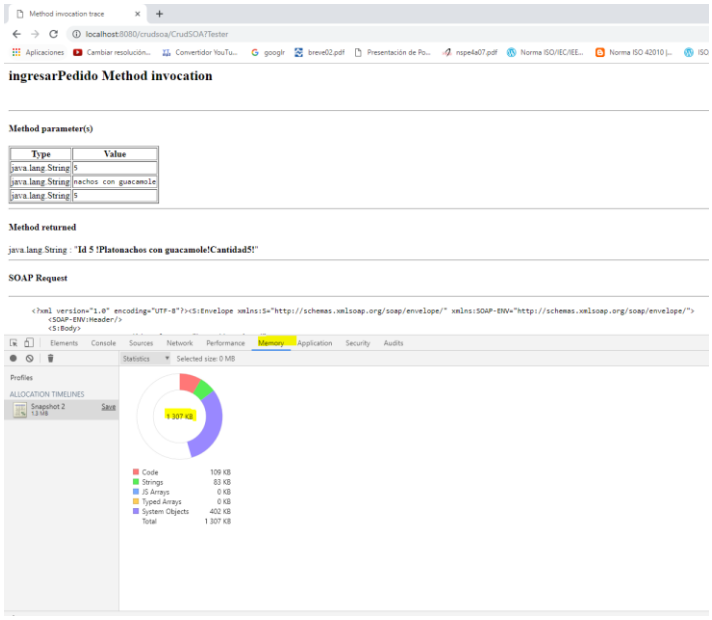



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**63. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 5897ms como se muestra en la Figura 33:

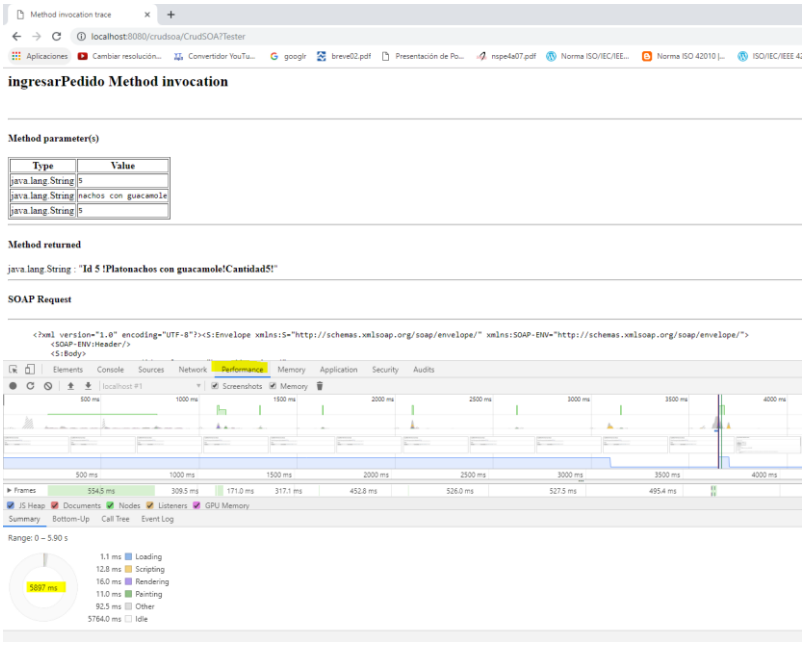



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 6:**

**64. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 24ms como se muestra en la Figura 34:

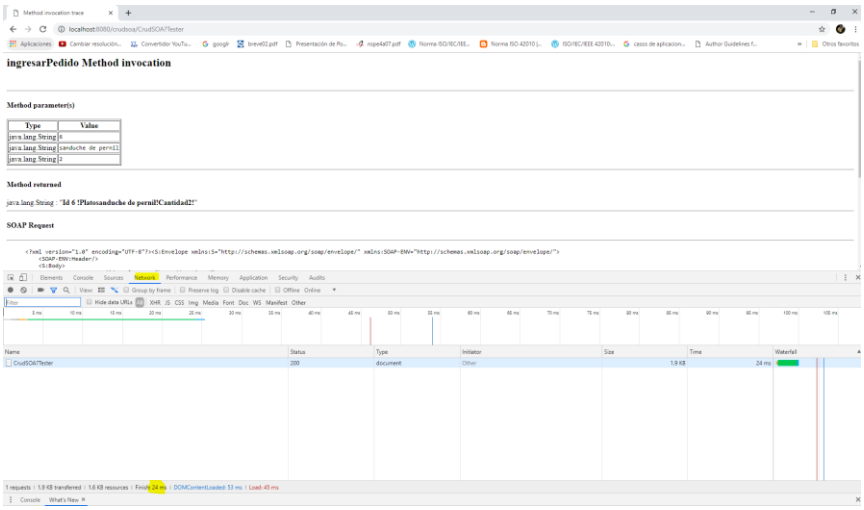



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**65. Memoria:** Al hacer la siguiente iteración consume 1311 kb como se muestra en la Figura 32.

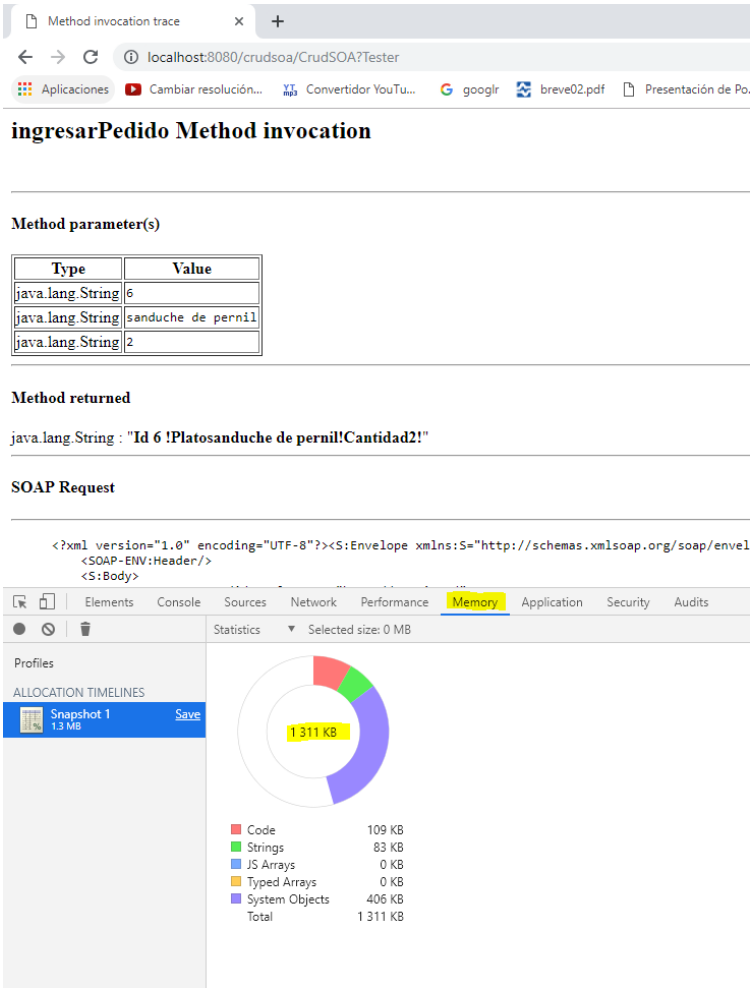



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**66. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 4097ms como se muestra en la Figura 33:

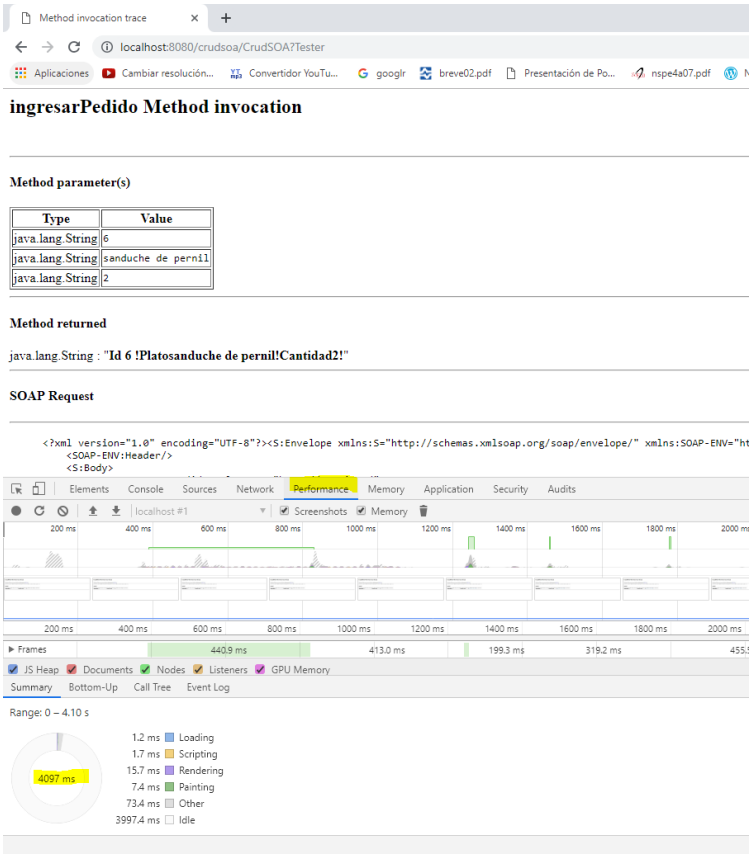



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 7:**

**67. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 20ms como se muestra en la Figura 34:

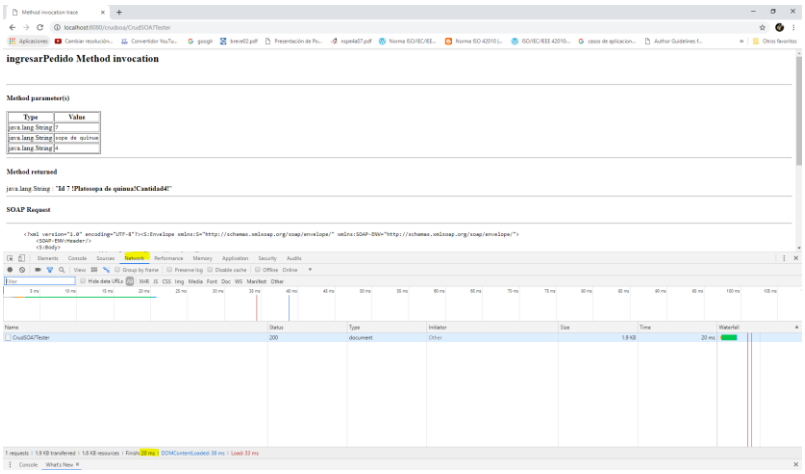


**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**68. Memoria:** Al hacer la siguiente iteración consume 1311 kb como se muestra en la Figura 32.

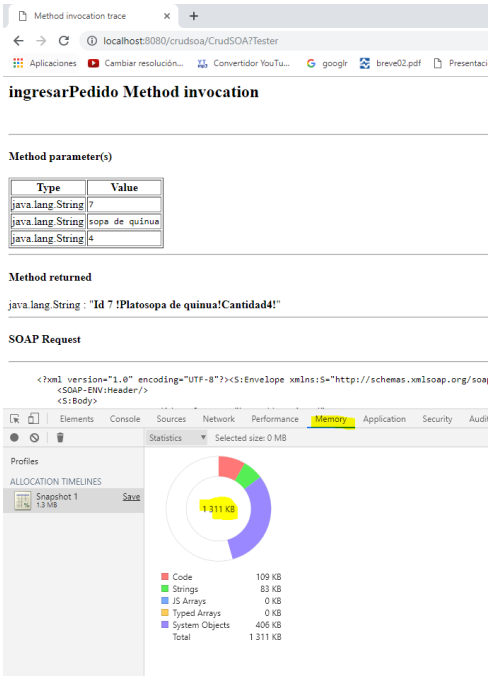



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**69. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 5253ms como se muestra en la Figura 33:

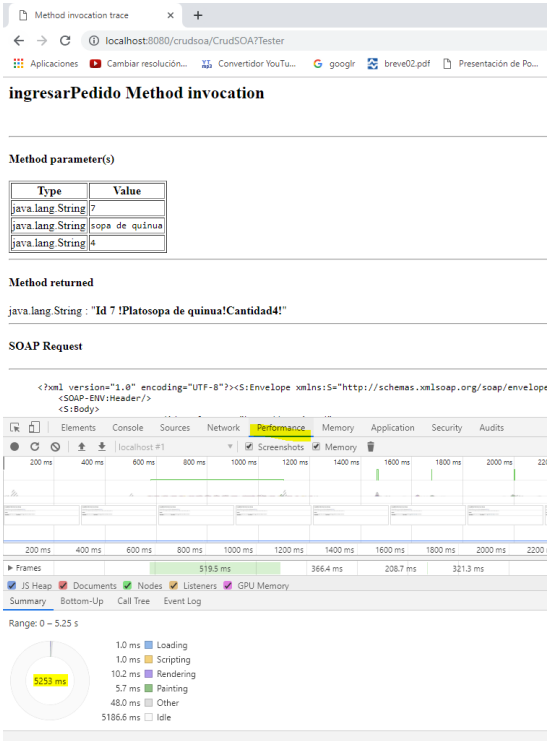



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 8:**

**70. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 19ms como se muestra en la Figura 34:

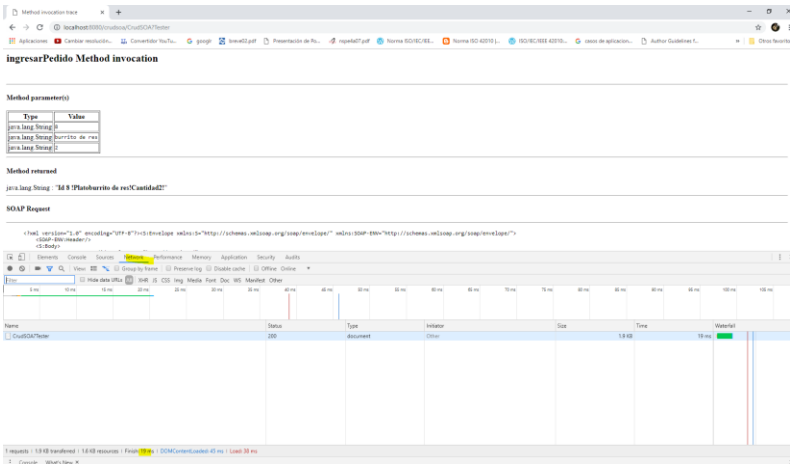



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**71. Memoria:** Al hacer la siguiente iteración consume 1317 kb como se muestra en la Figura 32.

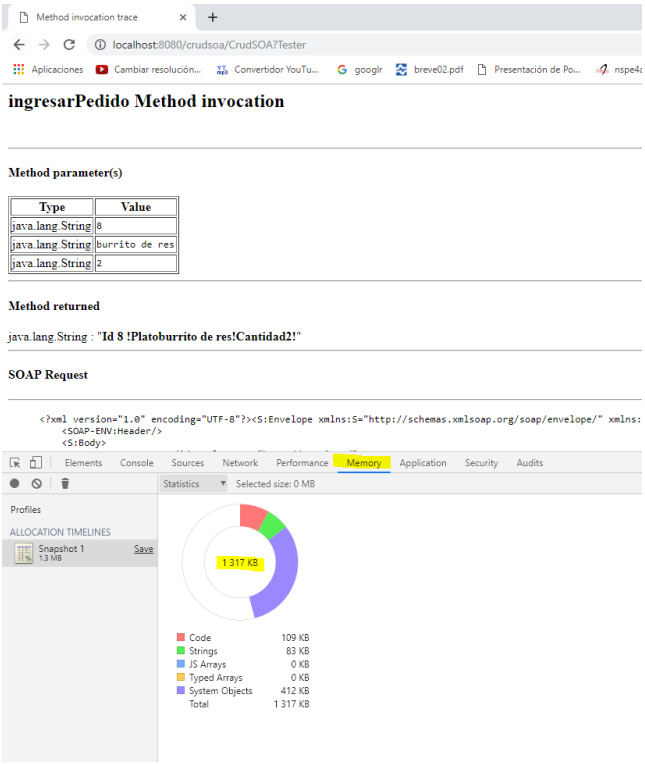



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**72. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 6304ms como se muestra en la Figura 33:

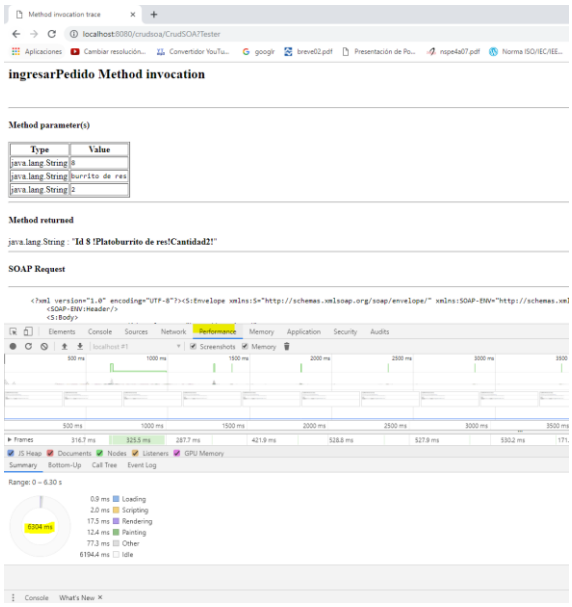



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 9:**

**73. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 22ms como se muestra en la Figura 34:

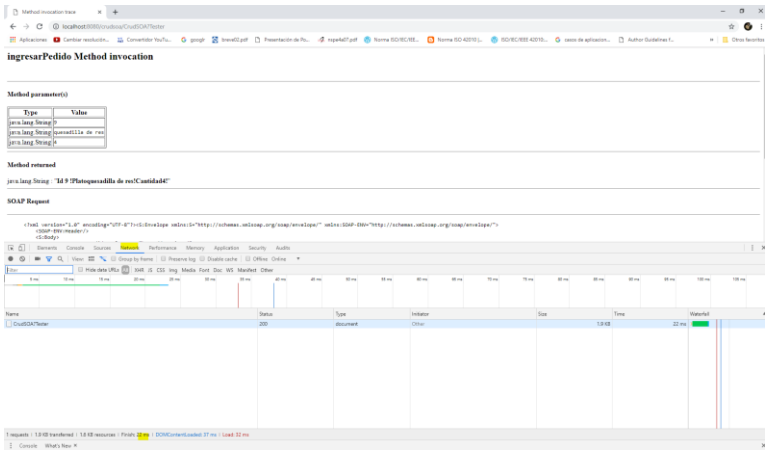



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**74. Memoria:** Al hacer la siguiente iteración consume 1317 kb como se muestra en la Figura 32.

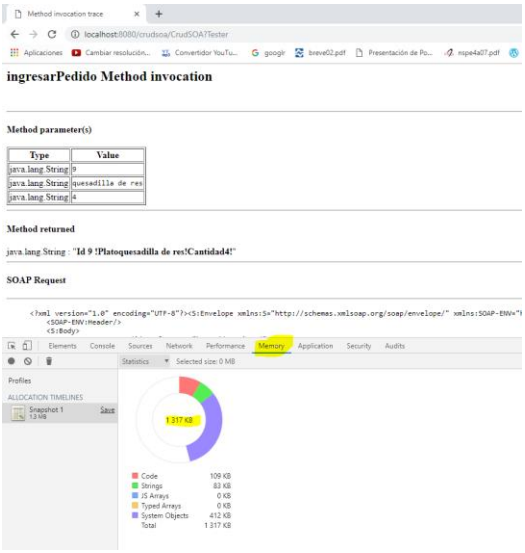



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

75. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 6304ms como se muestra en la Figura 33:

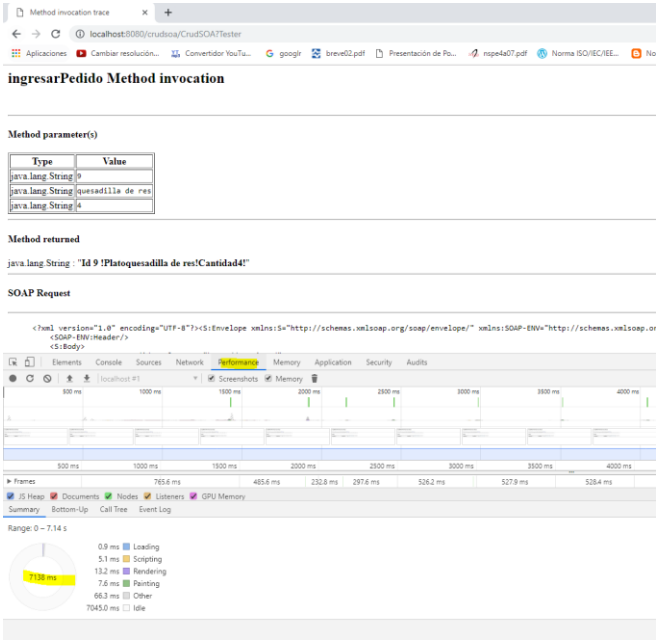



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 10:**

76. **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 28ms como se muestra en la Figura 34:

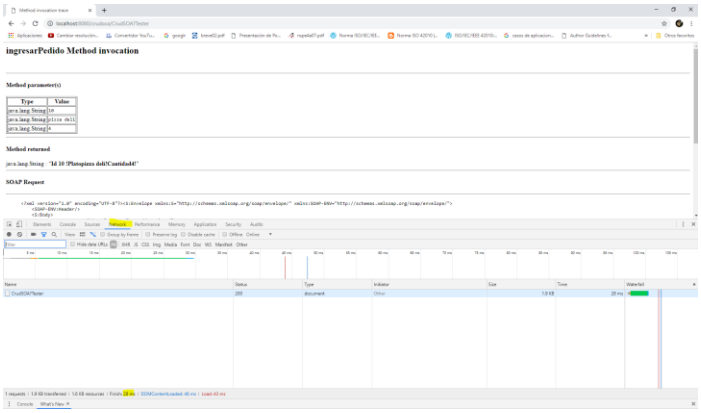


**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

77. **Memoria:** Al hacer la siguiente iteración consume 1326 kb como se muestra en la Figura 32.

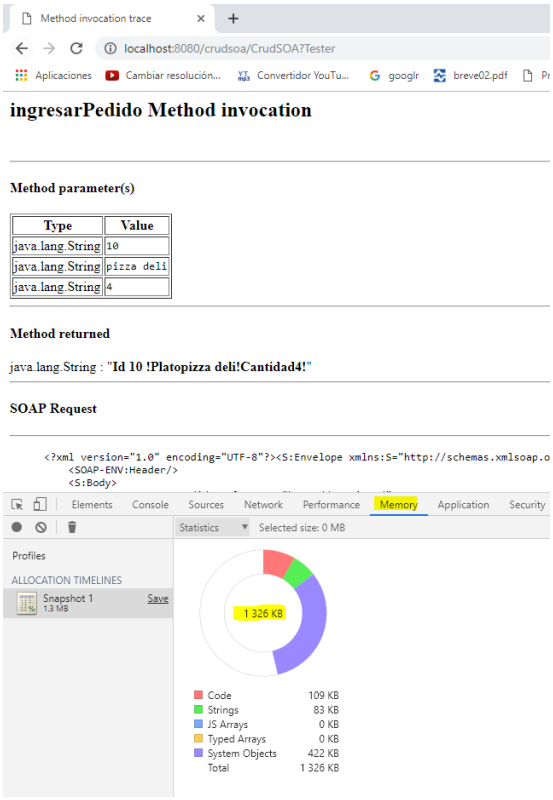



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

78. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 8274ms como se muestra en la Figura 33:

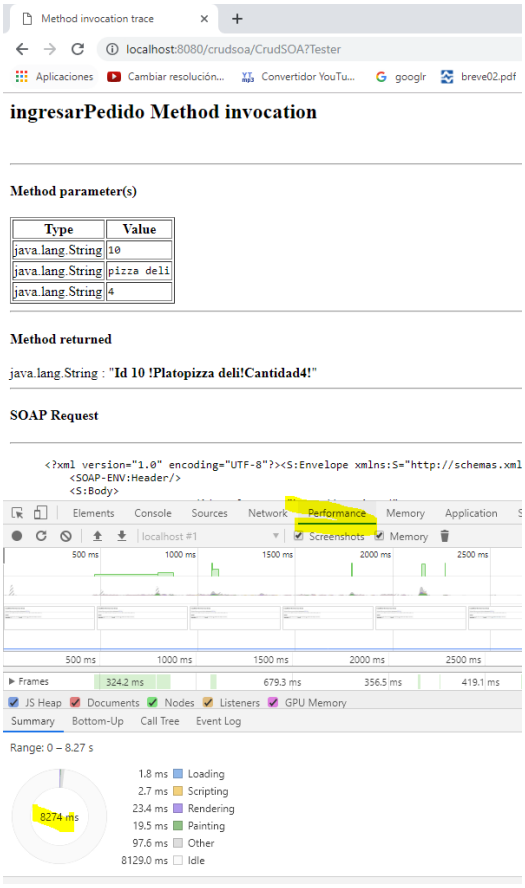



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 11:**

**79. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 22ms como se muestra en la Figura 34:

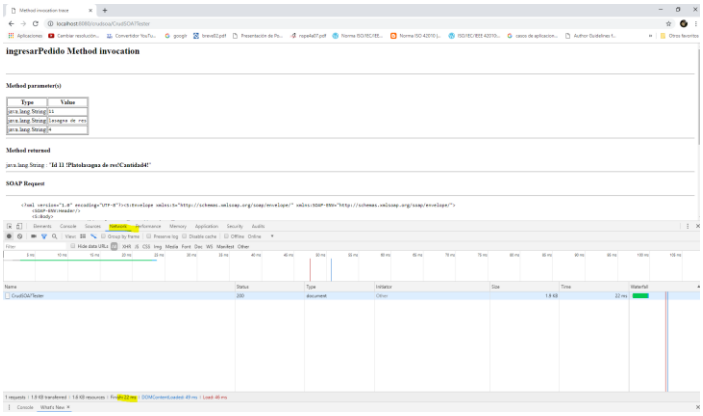


**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**80. Memoria:** Al hacer la siguiente iteración consume 1326 kb como se muestra en la Figura 32.

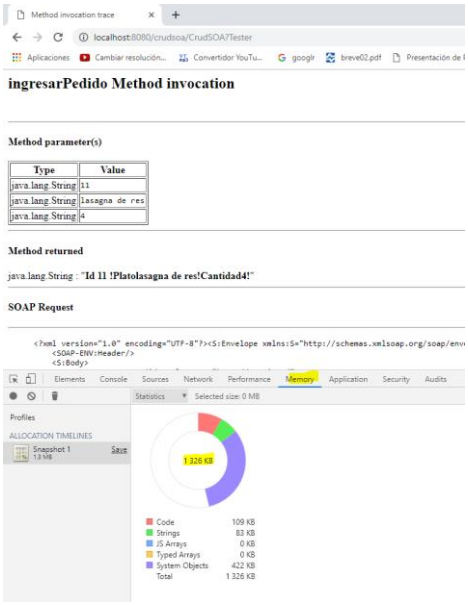



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**81. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 7944ms como se muestra en la Figura 33:

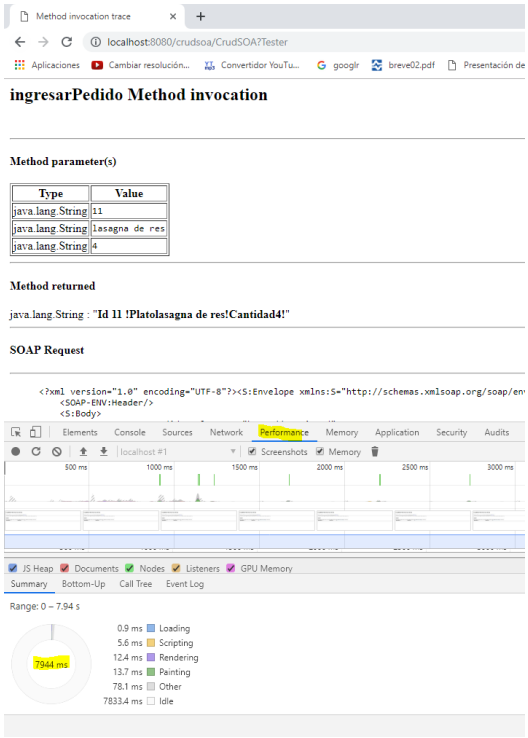



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 12:**

**82. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 22ms como se muestra en la Figura 34:

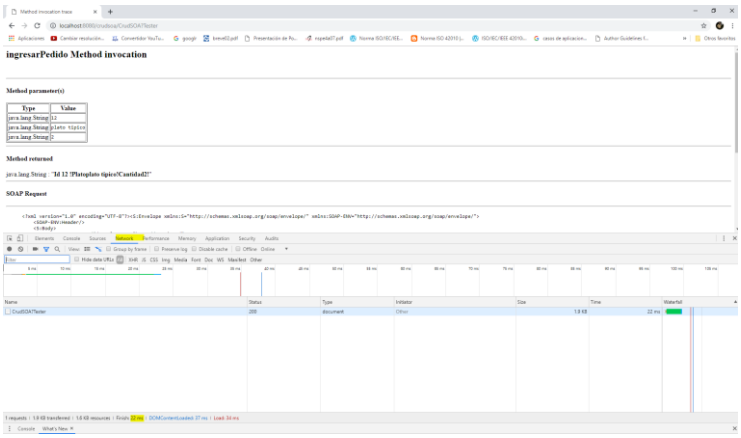



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**83. Memoria:** Al hacer la siguiente iteración consume 1326 kb como se muestra en la Figura 32.

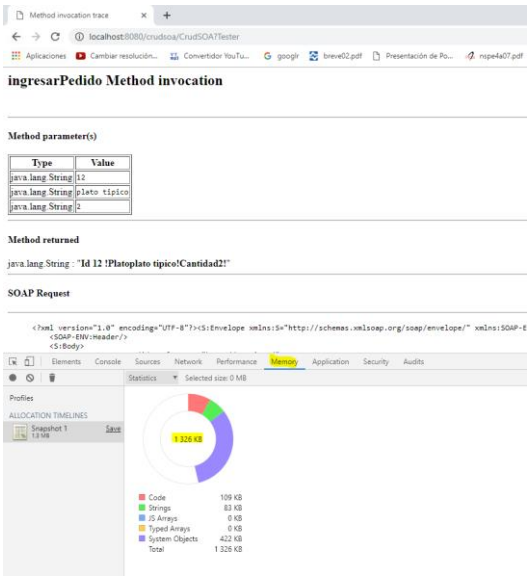



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**84. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 8266ms como se muestra en la Figura 33:

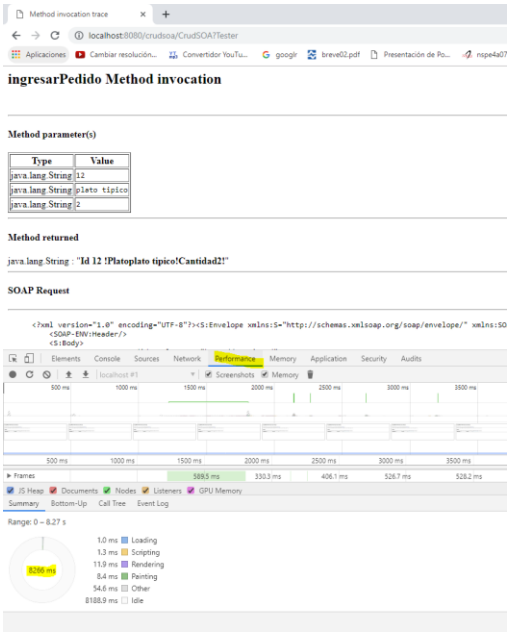



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 13:**

**85. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 14ms como se muestra en la Figura 34:

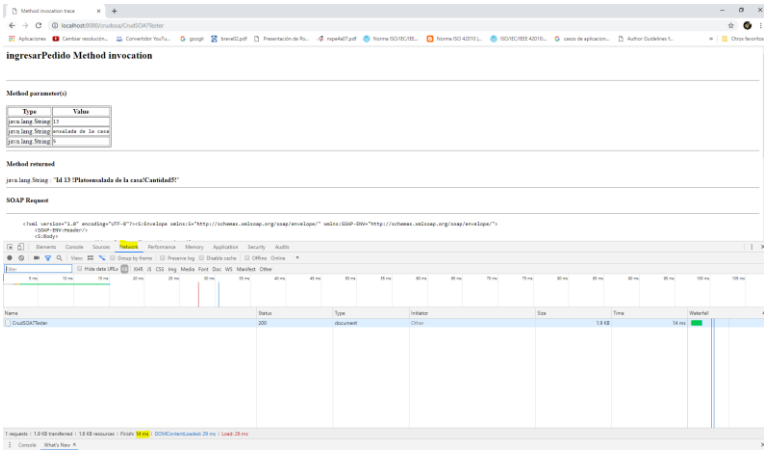



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**86. Memoria:** Al hacer la siguiente iteración consume 1326 kb como se muestra en la Figura 32.

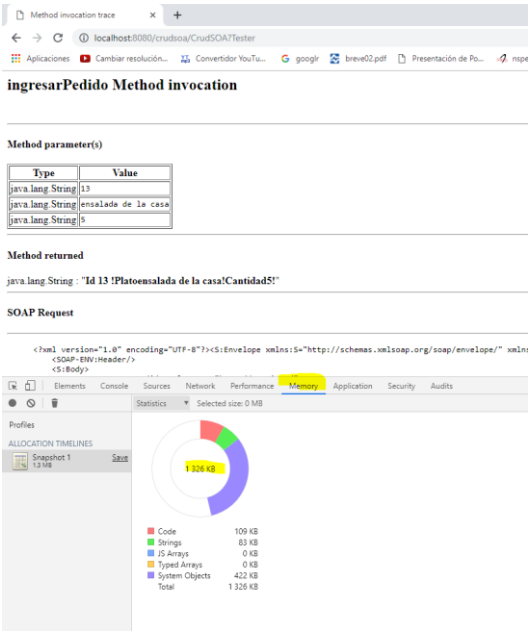



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

87. **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 7515ms como se muestra en la Figura 33:

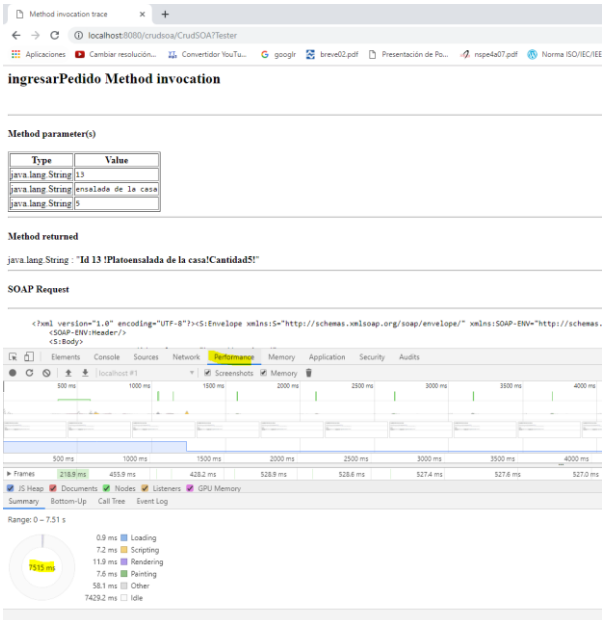



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 14:**

88. **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 17ms como se muestra en la Figura 34:

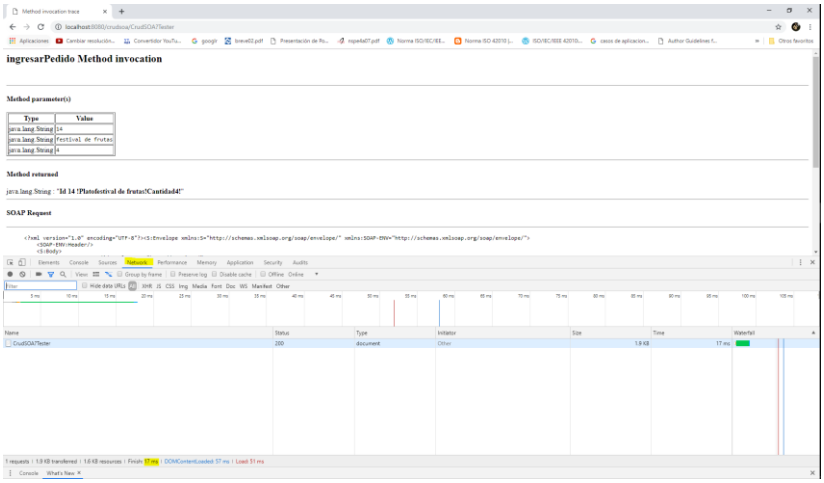



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**89. Memoria:** Al hacer la siguiente iteración consume 1326 kb como se muestra en la Figura 32.

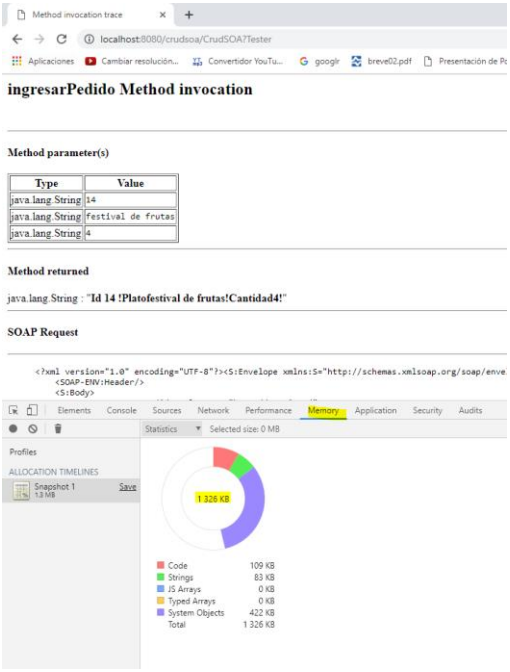



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**90. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 7530ms como se muestra en la Figura 33:

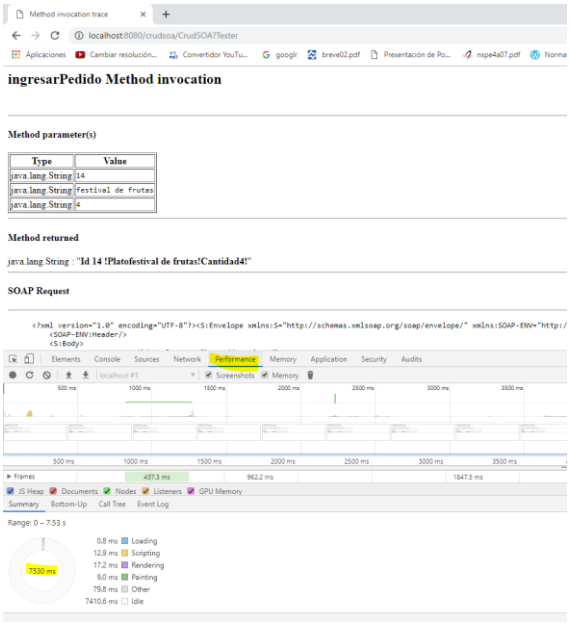



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

**Fuente:** Propia

- **Iteración nro. 15:**

**91. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 24ms como se muestra en la Figura 34:

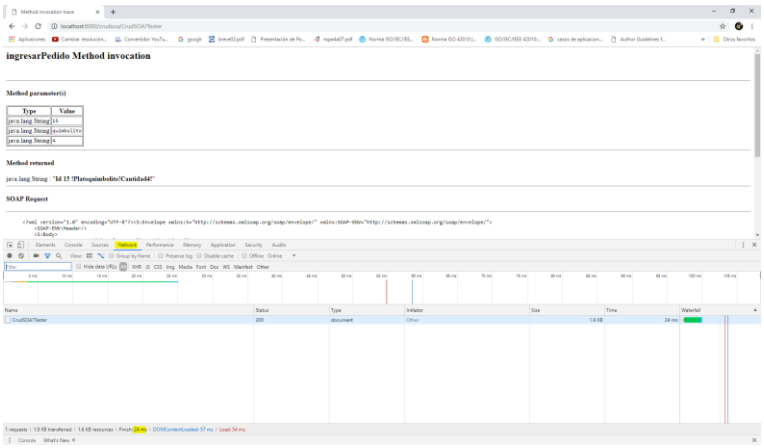



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**92. Memoria:** Al hacer la siguiente iteración consume 1340 kb como se muestra en la Figura 32.

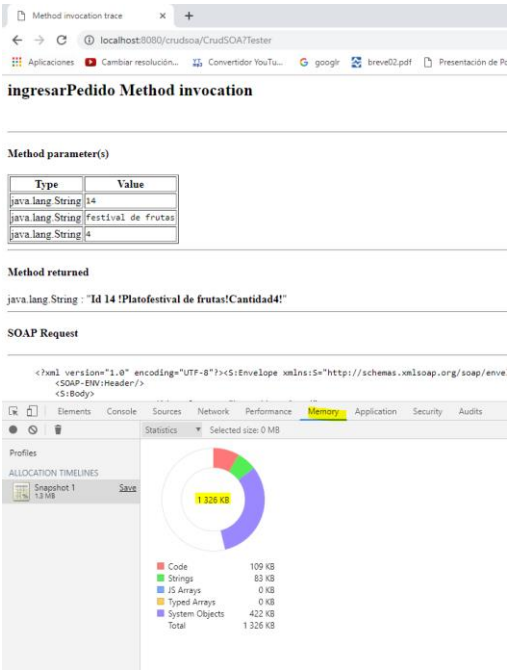



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**93. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 6504ms como se muestra en la Figura 33:

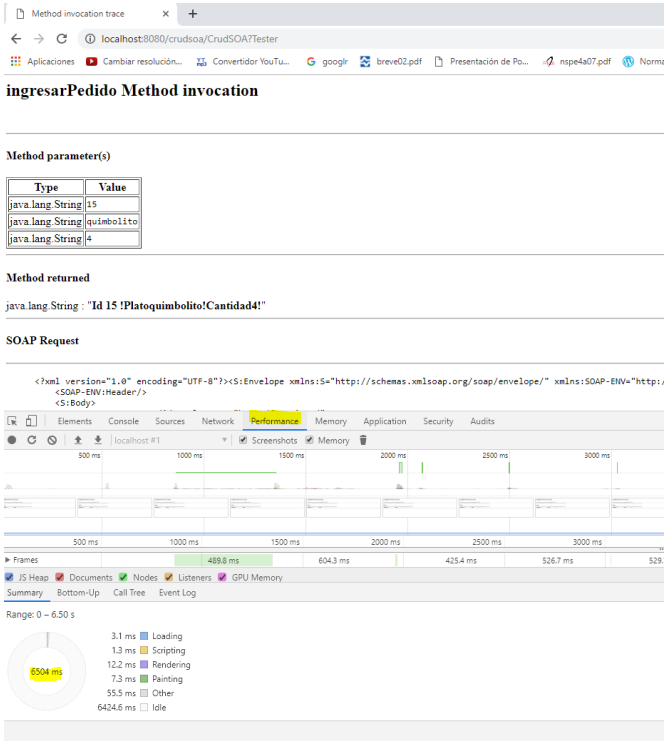



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 16:**

**94. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 16ms como se muestra en la Figura 34:

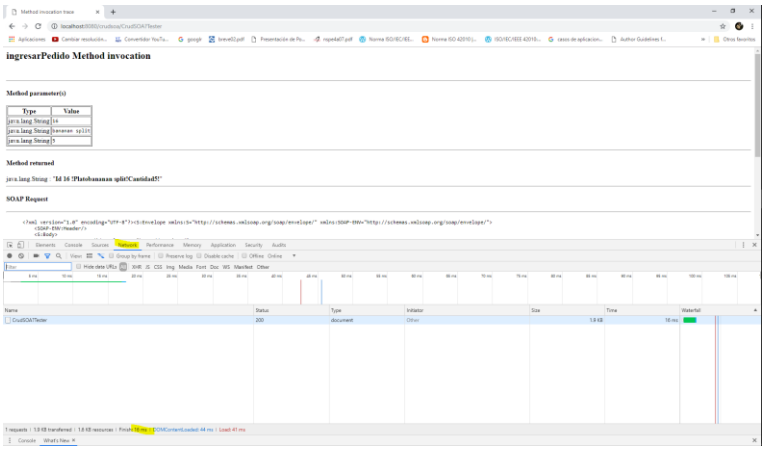



**Fig. 37.** Arquitectura SOA-Red
**Fuente:** Propia

**95. Memoria:** Al hacer la siguiente iteración consume 1340 kb como se muestra en la Figura 32.

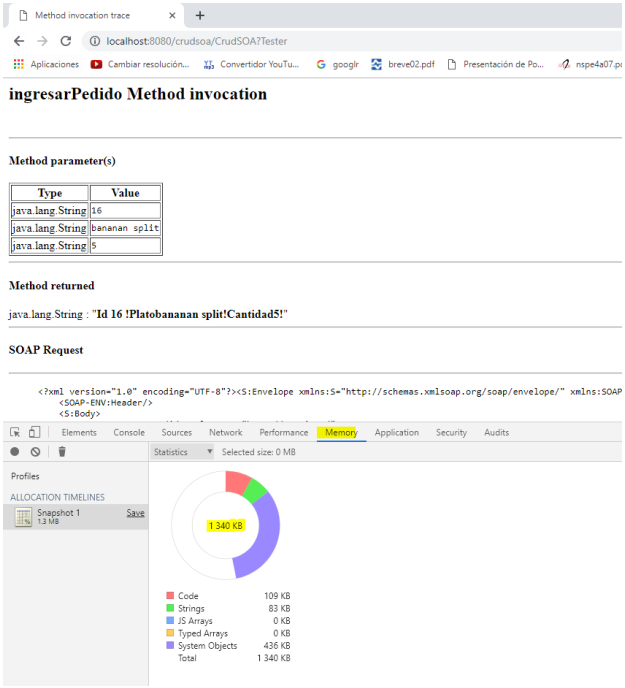



**Fig. 38.** Arquitectura SOA-Memoria
**Fuente:** Propia

**96. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 7626ms como se muestra en la Figura 33:

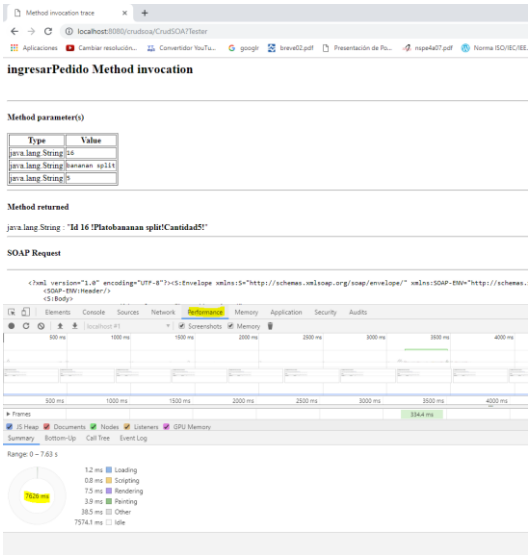



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 17:**

**97. Red:** Al hacerla la siguiente iteración se realiza en un tiempo 15ms como se muestra en la Figura 34:

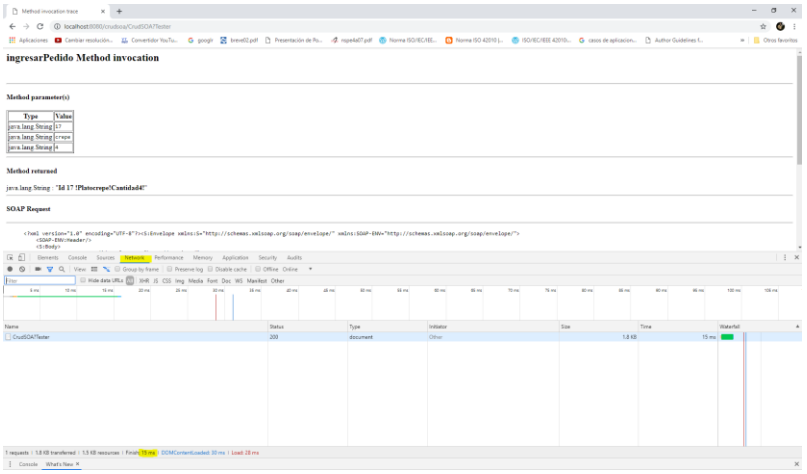



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**98. Memoria:** Al hacer la siguiente iteración consume 1847 kb como se muestra en la Figura 32.

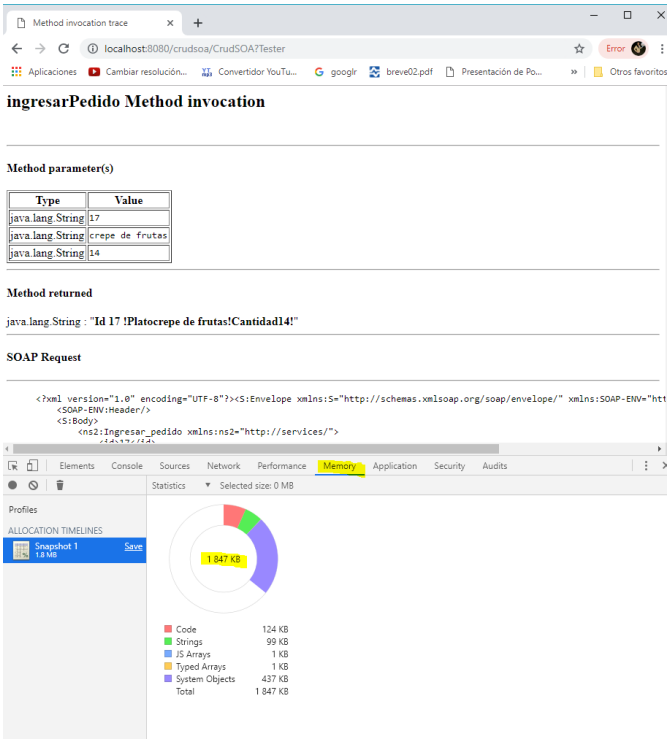



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**99. Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 7927ms como se muestra en la Figura 33:

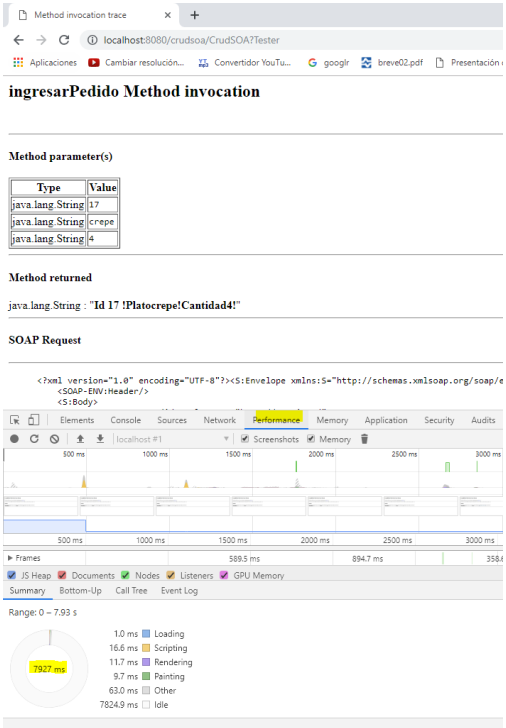



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 18:**

**100.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 18ms como se muestra en la Figura 34:

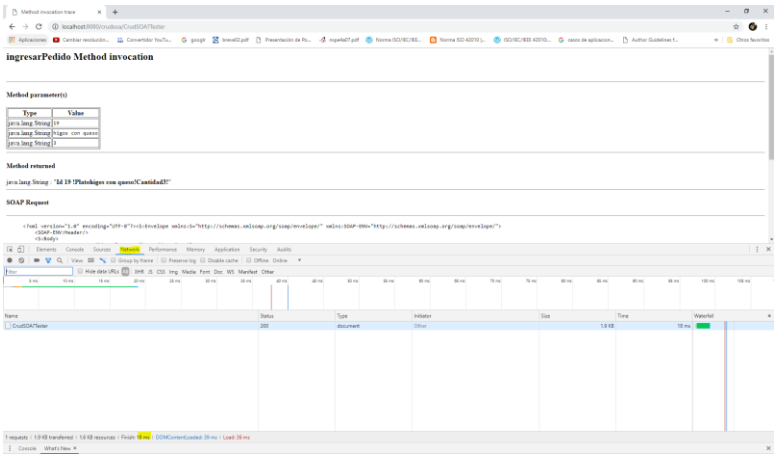



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**101.** **Memoria:** Al hacer la siguiente iteración consume 1847 kb como se muestra en la Figura 32.

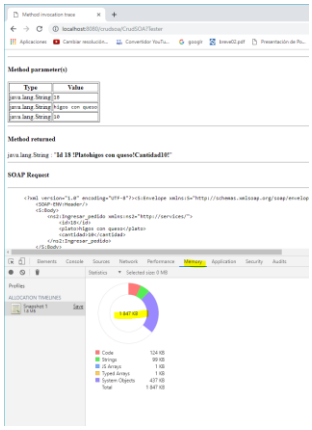



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**102.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 7771ms como se muestra en la Figura 33:

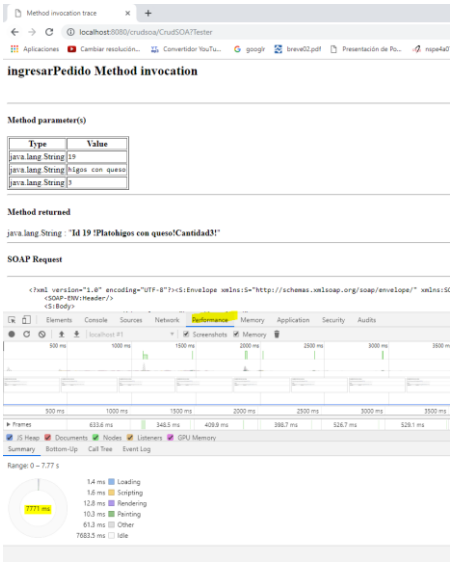



**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 19:**

**103.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 15ms como se muestra en la Figura 34:

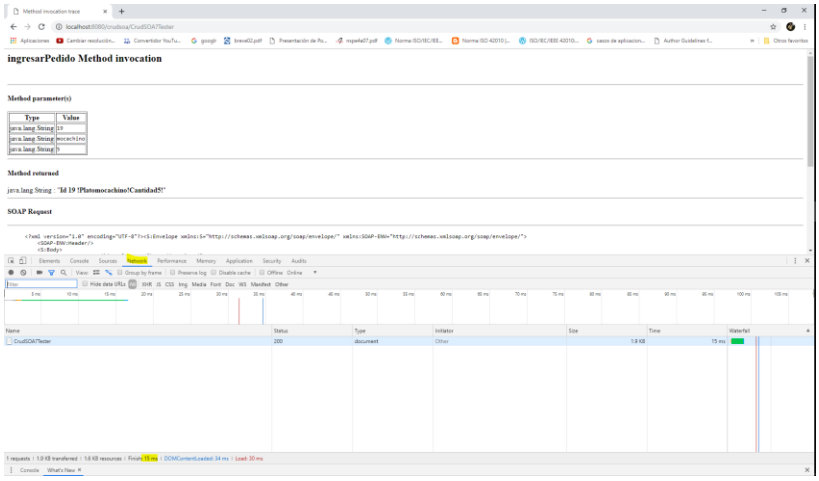



**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**104.** **Memoria:** Al hacer la siguiente iteración consume 1854 kb como se muestra en la Figura 32.

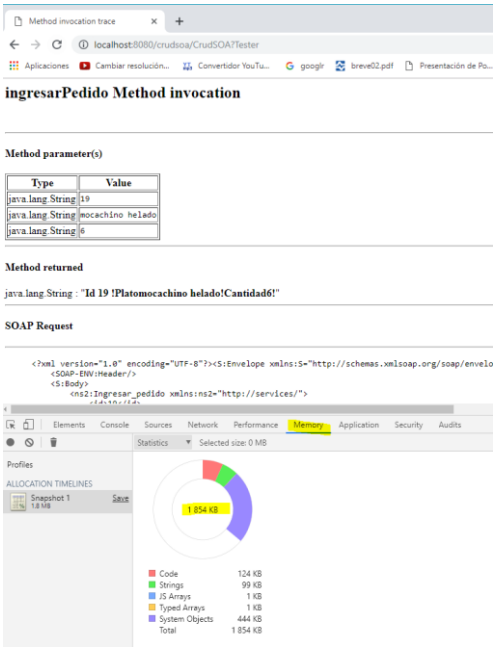



**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**105.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 7242ms como se muestra en la Figura 33:


**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

- **Iteración nro. 20:**

**106.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 23ms como se muestra en la Figura 34:

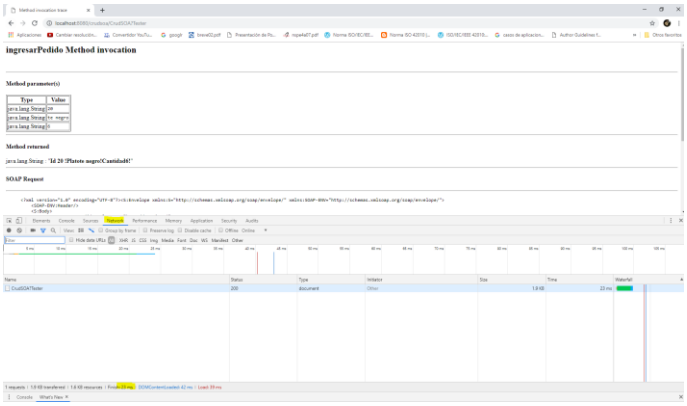


**Fig. 37.** Arquitectura SOA-Red

**Fuente:** Propia

**107.** **Memoria:** Al hacer la siguiente iteración consume 1854 kb como se muestra en la Figura 32.


**Fig. 38.** Arquitectura SOA-Memoria

**Fuente:** Propia

**108.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 7235ms como se muestra en la Figura 33:


**Fig. 39.** Arquitectura SOA-rendimiento

**Fuente:** Propia

## c) App Shell

- **Iteración nro. 3:**

**109.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 1.05s como se muestra en la Figura 34:

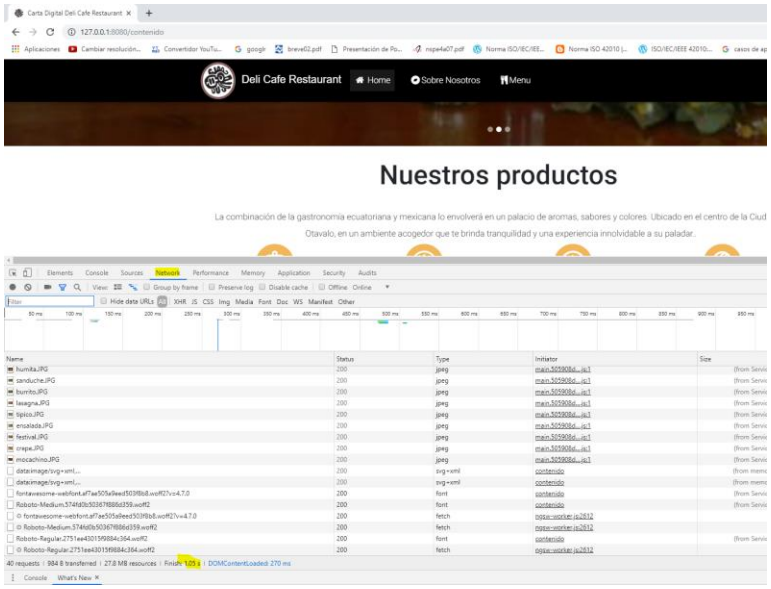



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**110.** **Memoria:** Al hacer la siguiente iteración consume 8691 kb como se muestra en la Figura 32.

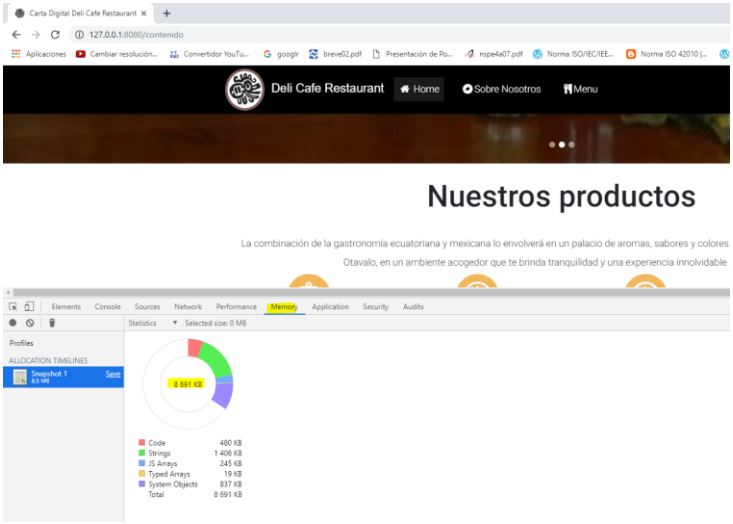



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**111.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1073 ms como se muestra en la Figura 33:

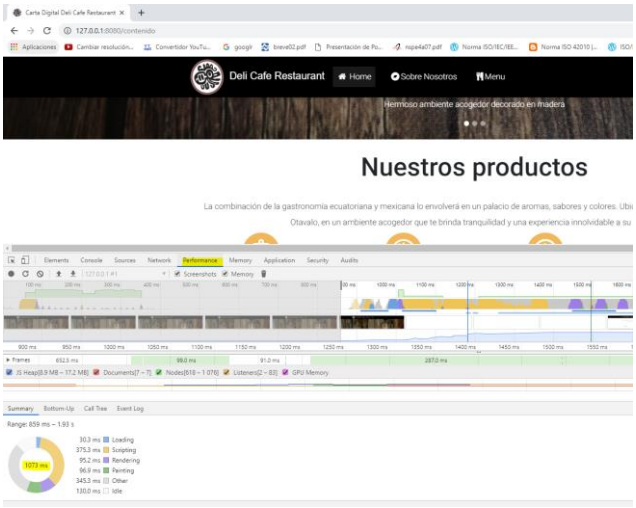



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 4:**

**112.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 1.20s como se muestra en la Figura 34:

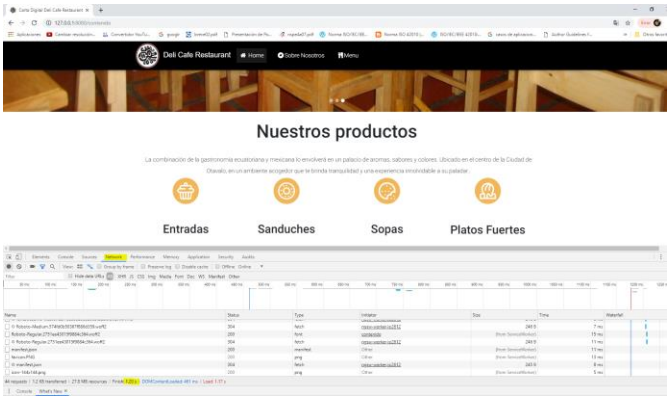



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**113.** **Memoria:** Al hacer la siguiente iteración consume1608 kb como se muestra en la Figura 32.

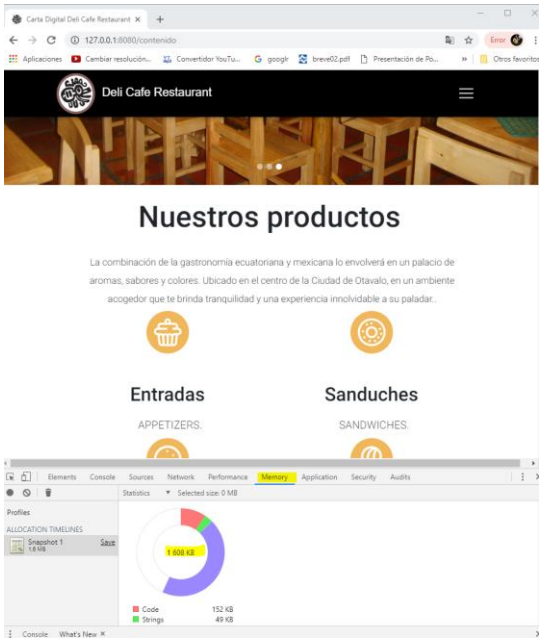



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**114.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1249 ms como se muestra en la Figura 33:

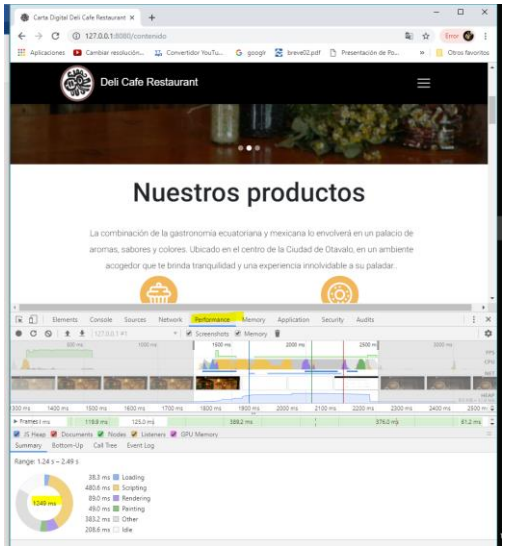



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 5:**

**115.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 1.01s como se muestra en la Figura 34:

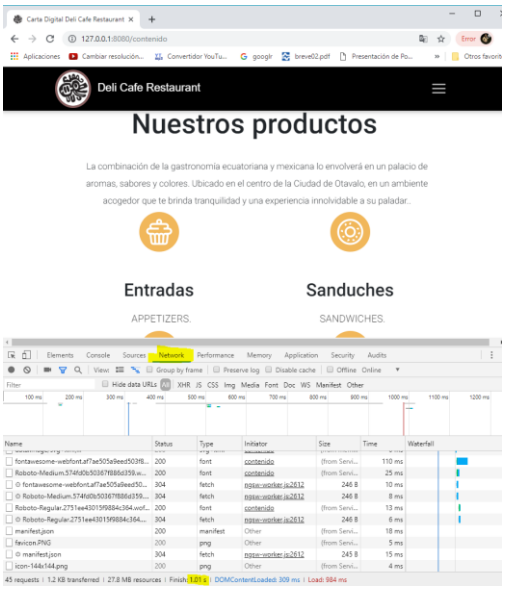



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**116.** **Memoria:** Al hacer la siguiente iteración consume1634 kb como se muestra en la Figura 32.

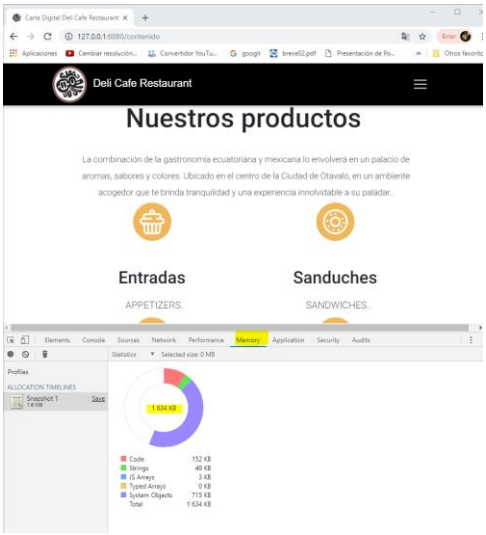



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**117.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1171ms como se muestra en la Figura 33:

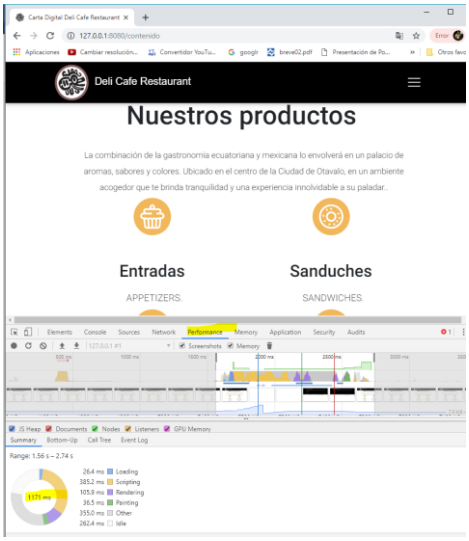



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 6:**

**118.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 1.15s como se muestra en la Figura 34:

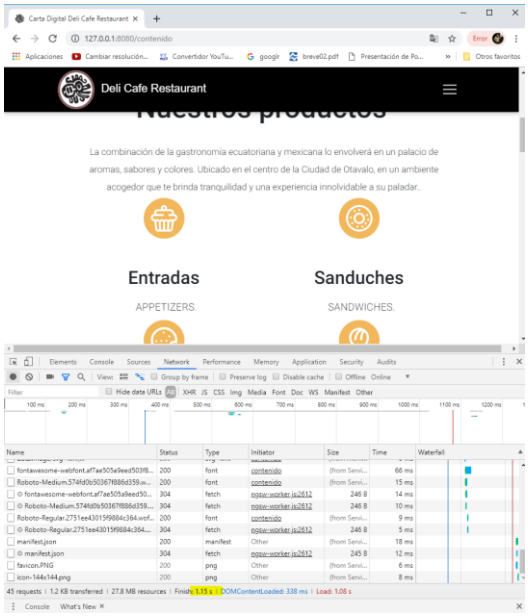



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**119.** **Memoria:** Al hacer la siguiente iteración consume1655 kb como se muestra en la Figura 32.

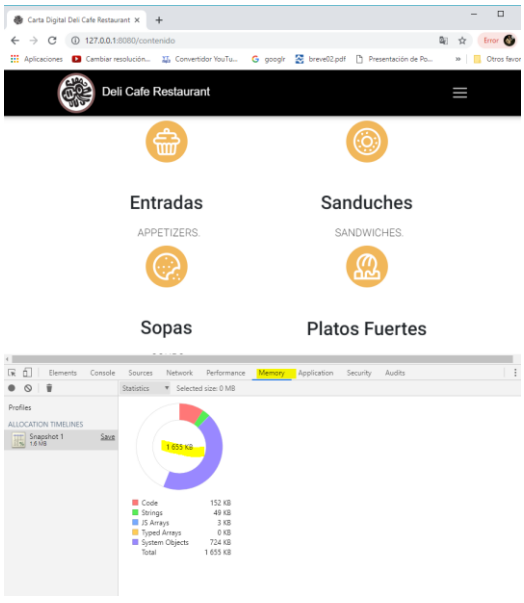



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**120.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1062ms como se muestra en la Figura 33:

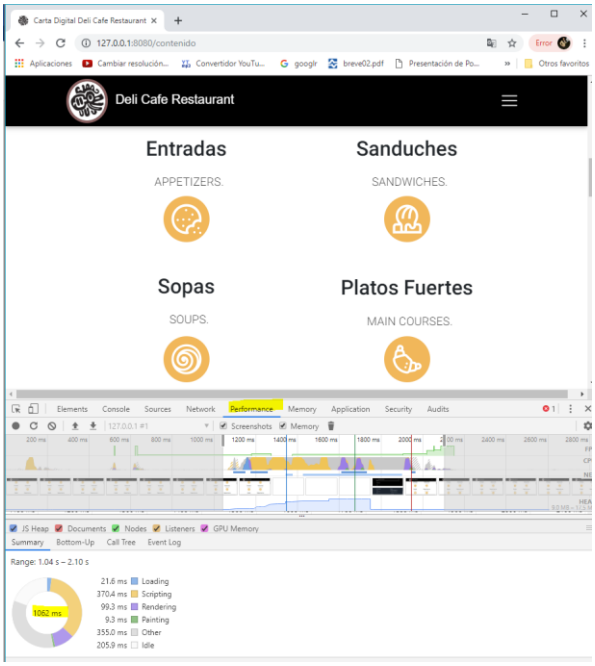



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 7:**

**121.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 1.17s como se muestra en la Figura 34:

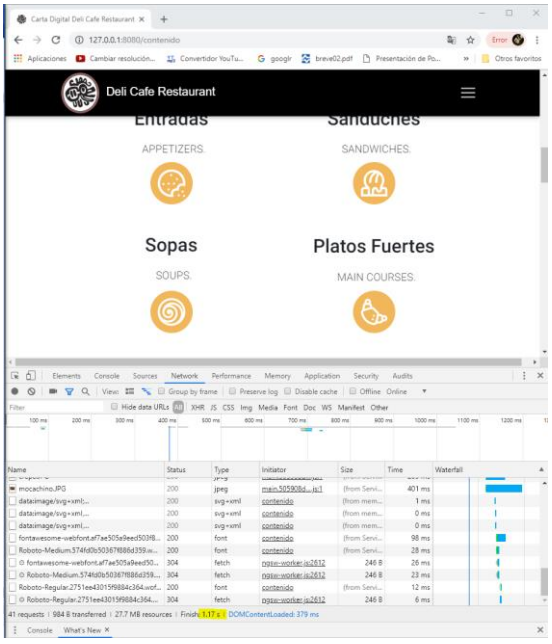



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**122.** **Memoria:** Al hacer la siguiente iteración consume1671 kb como se muestra en la Figura 32.

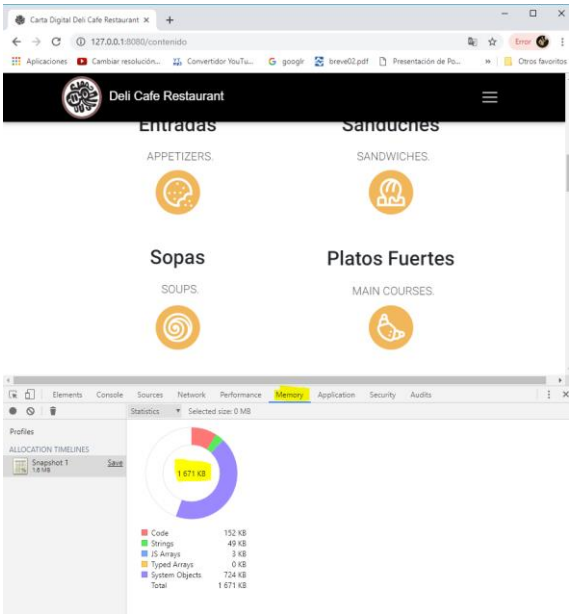



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**123.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 1090ms como se muestra en la Figura 33:

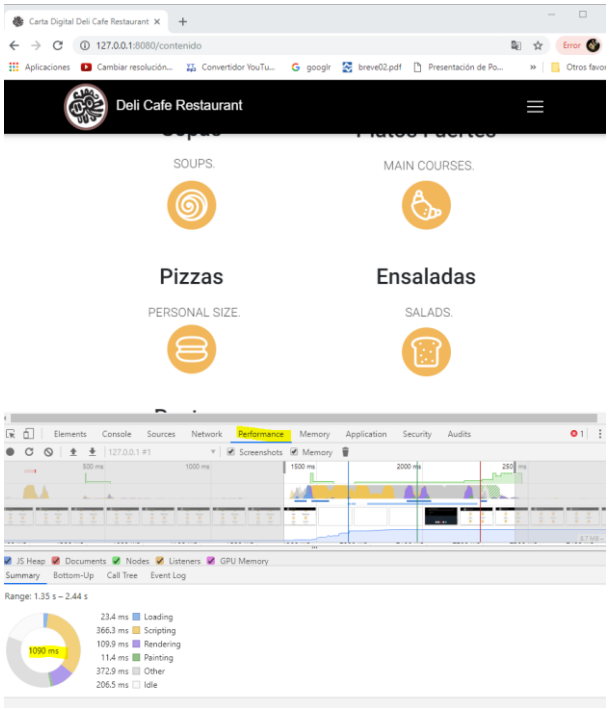



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 8:**

**124.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 1.28s como se muestra en la Figura 34:

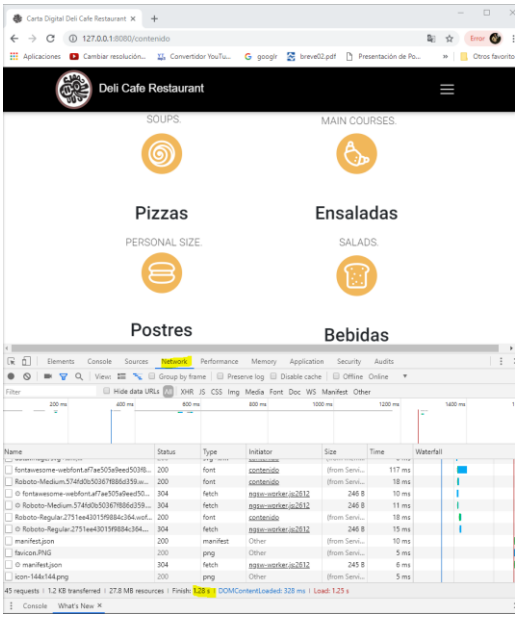



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**125.** **Memoria:** Al hacer la siguiente iteración consume1644 kb como se muestra en la Figura 32.

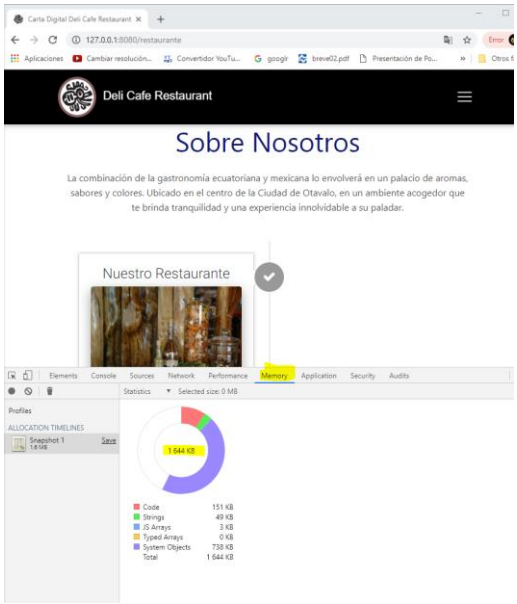



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**126.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 699ms como se muestra en la Figura 33:

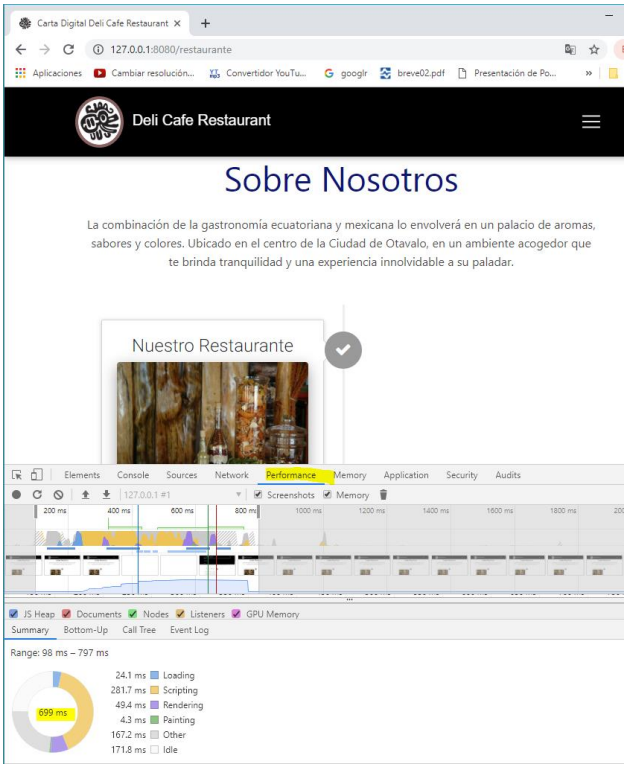



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 9:**

**127.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 566ms como se muestra en la Figura 34:

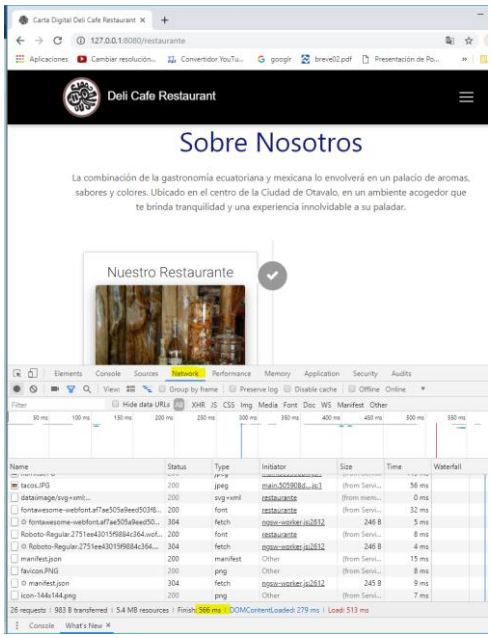



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**128.** **Memoria:** Al hacer la siguiente iteración consume 1675 kb como se muestra en la Figura 32.


**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**129.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 758ms como se muestra en la Figura 33:

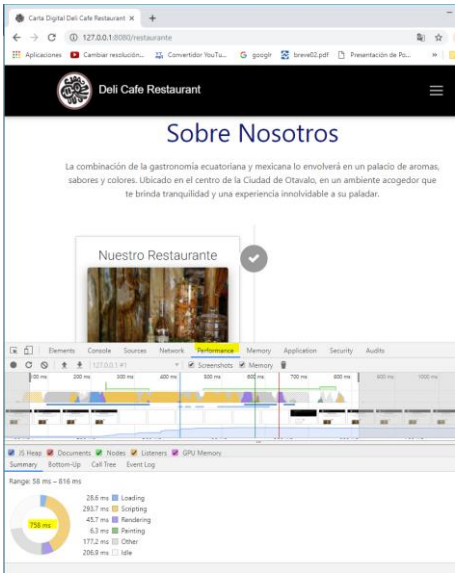



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 10:**

**130.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 756ms como se muestra en la Figura 34:

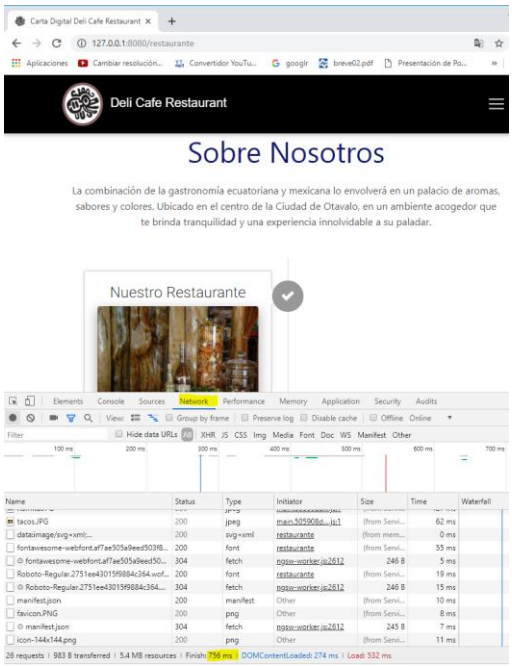



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**131.** **Memoria:** Al hacer la siguiente iteración consume 1675 kb como se muestra en la Figura 32.

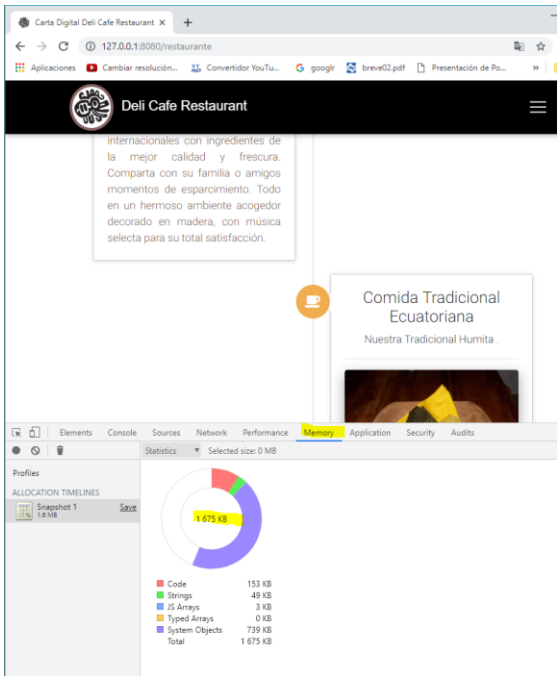



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**132.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 936ms como se muestra en la Figura 33:

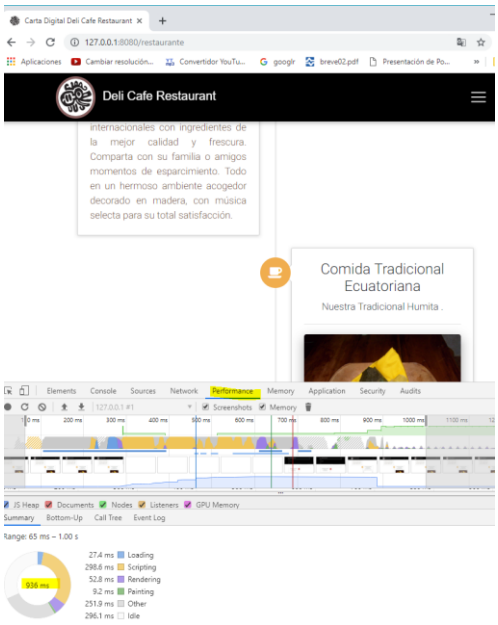



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 11:**

**133.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 689ms como se muestra en la Figura 34:

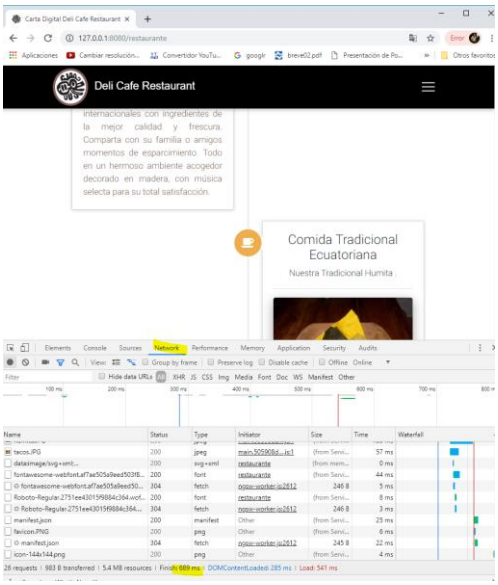



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**134.** **Memoria:** Al hacer la siguiente iteración consume1675 kb como se muestra en la Figura 32.

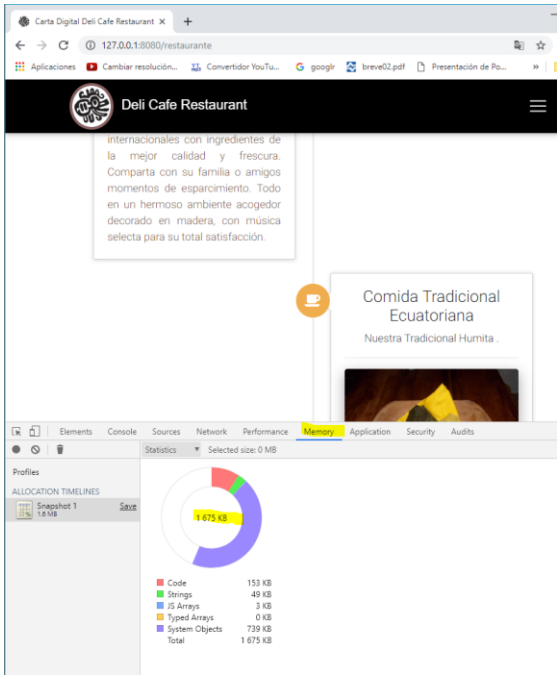



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**135.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 793ms como se muestra en la Figura 33:

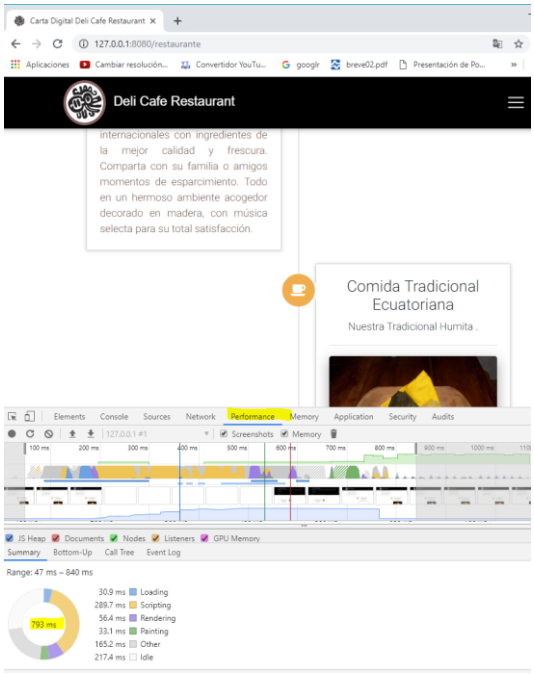



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 12:**

**136.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 585ms como se muestra en la Figura 34:

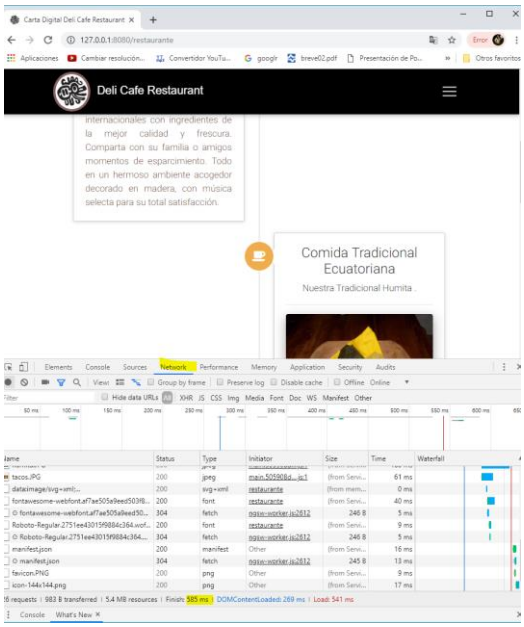



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**137.** **Memoria:** Al hacer la siguiente iteración consume1672 kb como se muestra en la Figura 32.

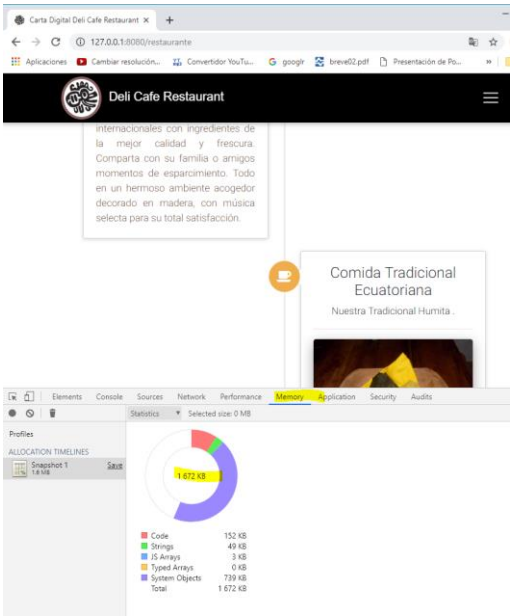



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**138.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 836ms como se muestra en la Figura 33:

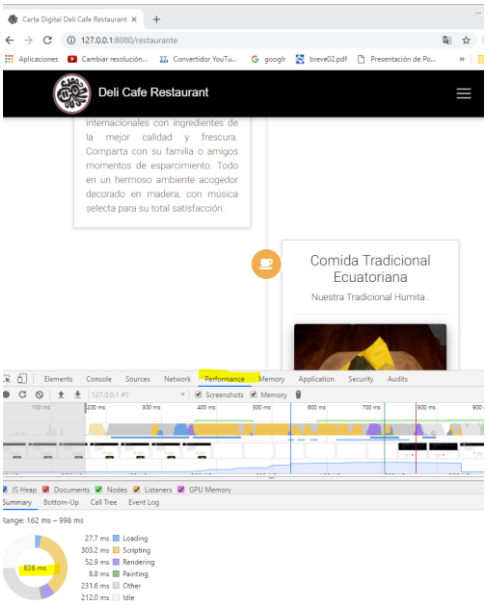



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 13:**

**139.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 565ms como se muestra en la Figura 34:

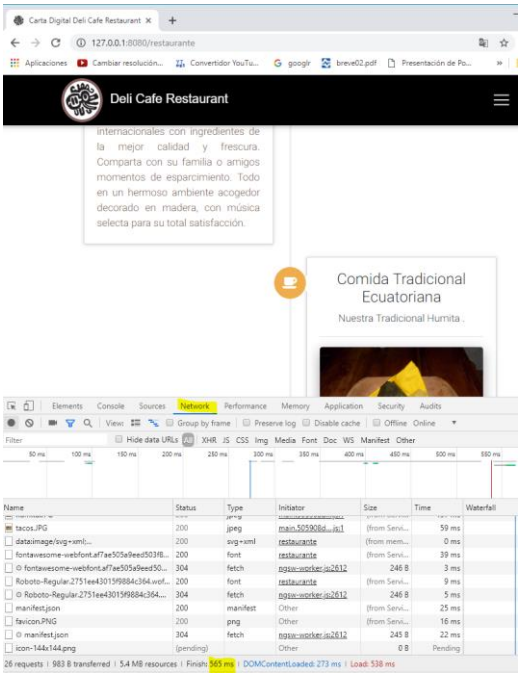



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**140.** **Memoria:** Al hacer la siguiente iteración consume1672 kb como se muestra en la Figura 32.

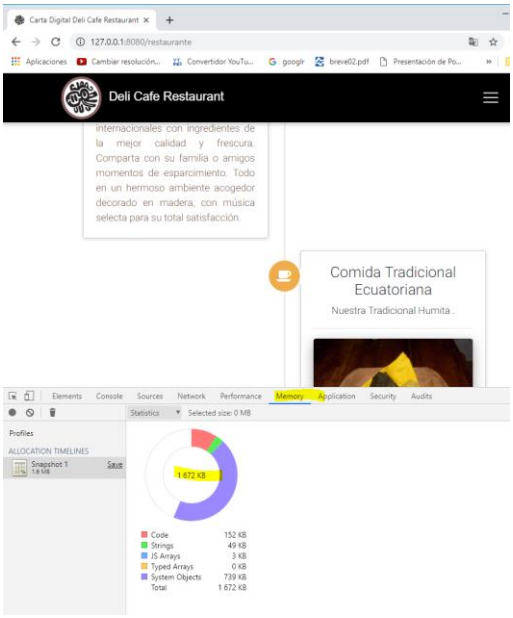



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**141.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 789ms como se muestra en la Figura 33:

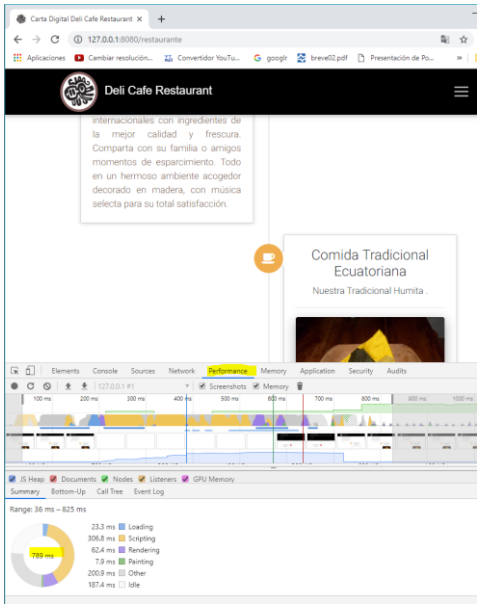



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 14:**

**142.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 761ms como se muestra en la Figura 34:

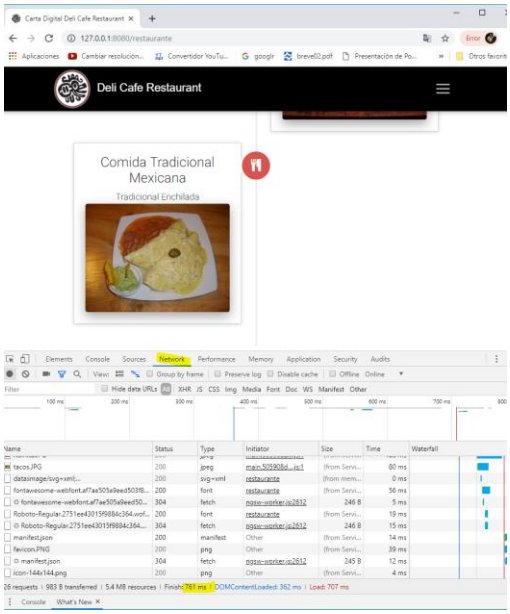



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**143.** **Memoria:** Al hacer la siguiente iteración consume 1730 kb como se muestra en la Figura 32.

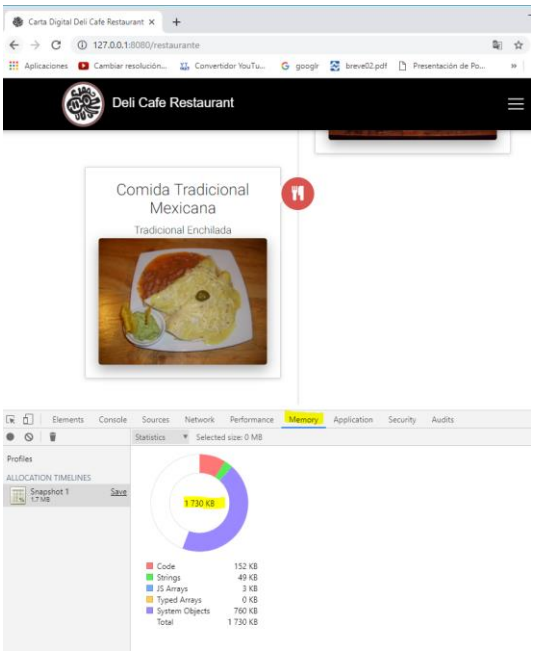



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**144.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 744ms como se muestra en la Figura 33:

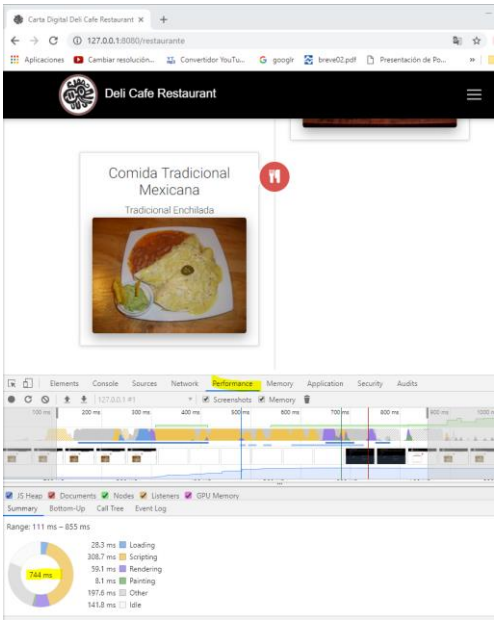



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 15:**

**145.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 761ms como se muestra en la Figura 34:

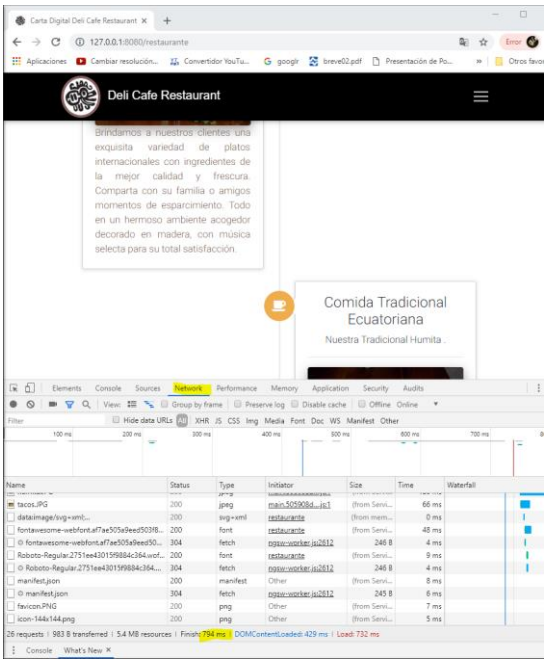



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**146.** **Memoria:** Al hacer la siguiente iteración consume 1734 kb como se muestra en la Figura 32.

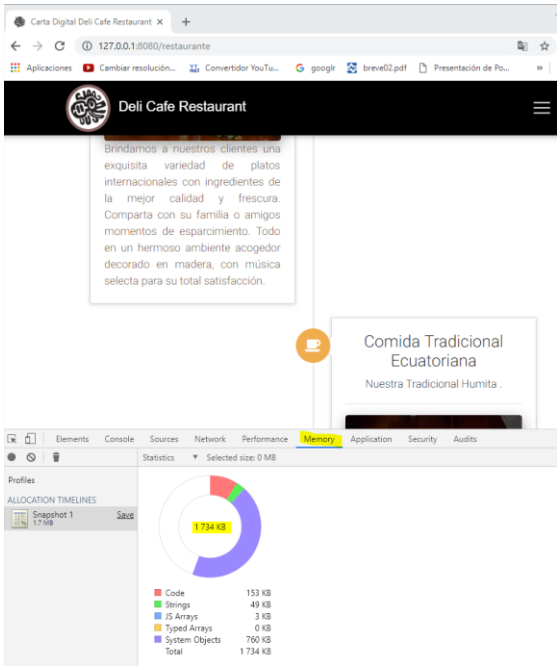



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**147.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 802ms como se muestra en la Figura 33:

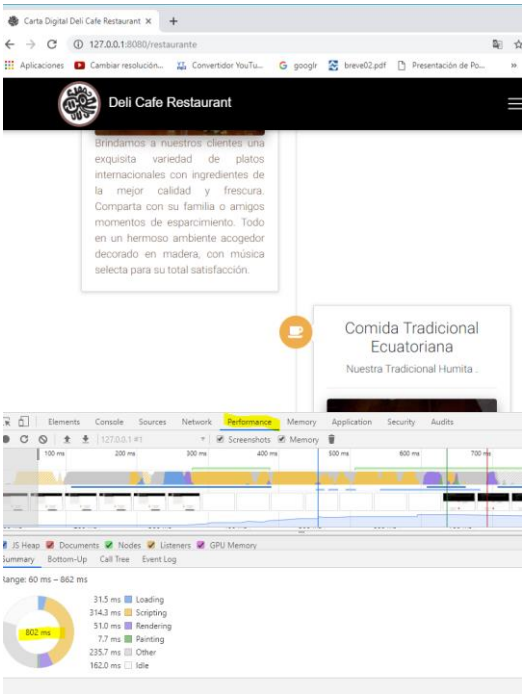



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 16:**

**148.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 718ms como se muestra en la Figura 34:

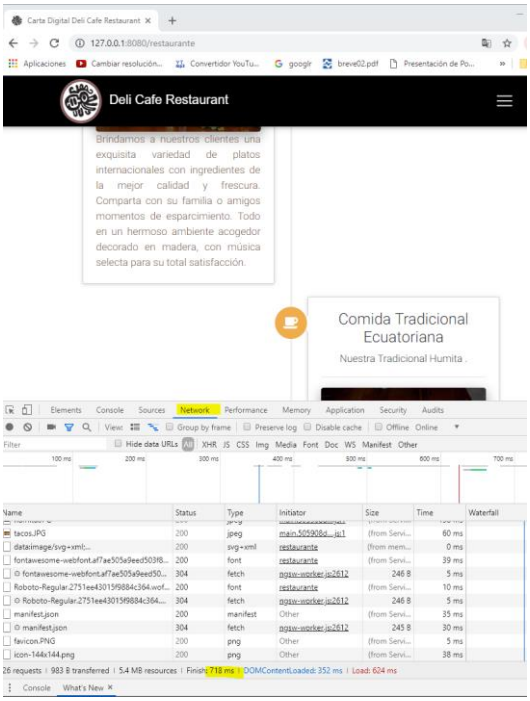



**Fig. 37.** Arquitectura App Shell-Red
**Fuente:** Propia

**149.** **Memoria:** Al hacer la siguiente iteración consume 1734 kb como se muestra en la Figura 32.

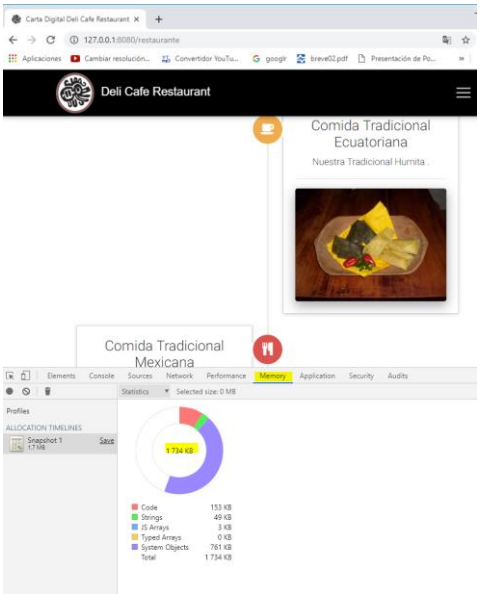



**Fig. 38.** Arquitectura App Shell-Memoria
**Fuente:** Propia

**150.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 325ms como se muestra en la Figura 33:

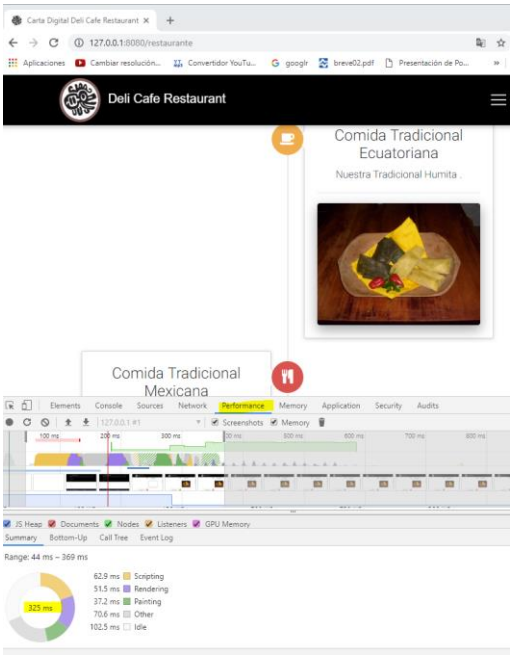



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 17:**

**151.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 566ms como se muestra en la Figura 34:

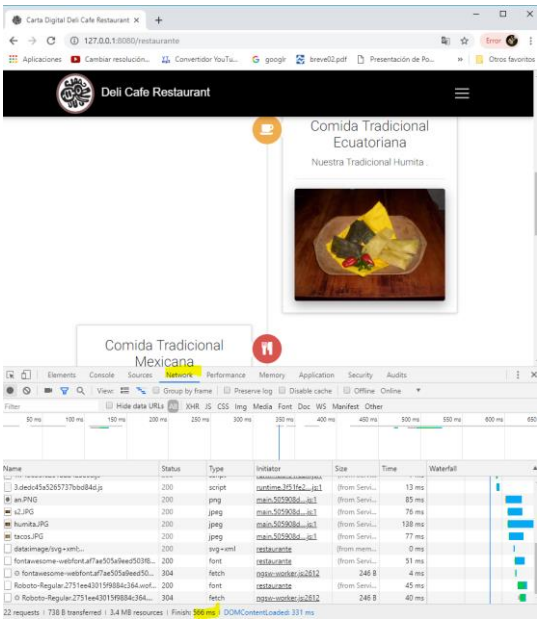



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**152.** **Memoria:** Al hacer la siguiente iteración consume1634 kb como se muestra en la Figura 32.

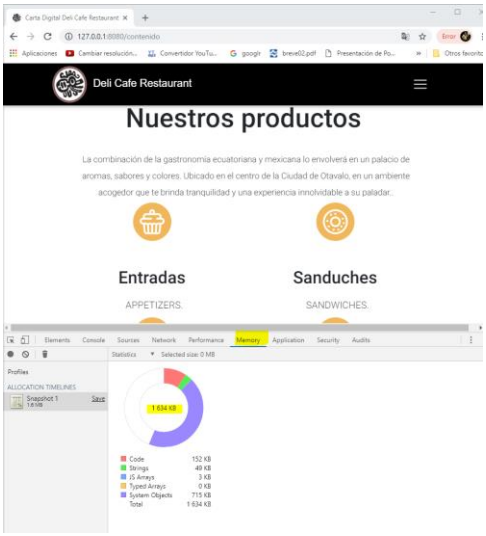



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**153.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 325ms como se muestra en la Figura 33:

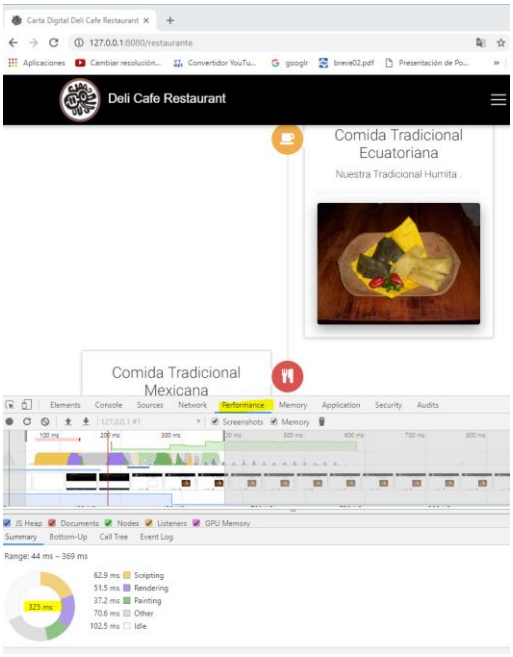



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 18:**

**154.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 718ms como se muestra en la Figura 34:

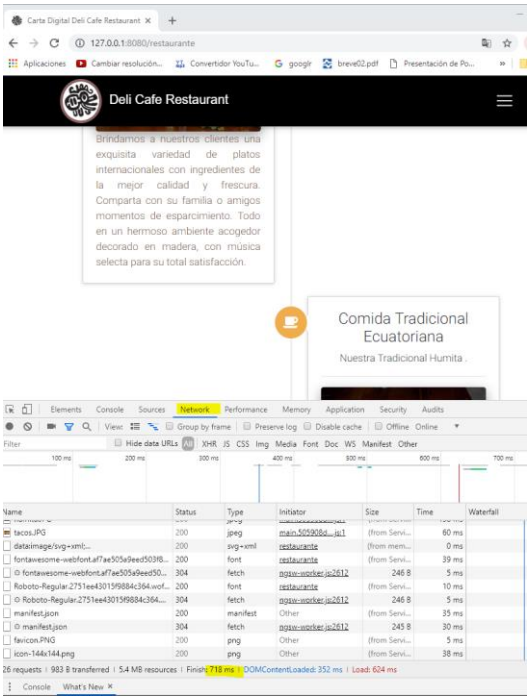



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**155.** **Memoria:** Al hacer la siguiente iteración consume 1634 kb como se muestra en la Figura 32.

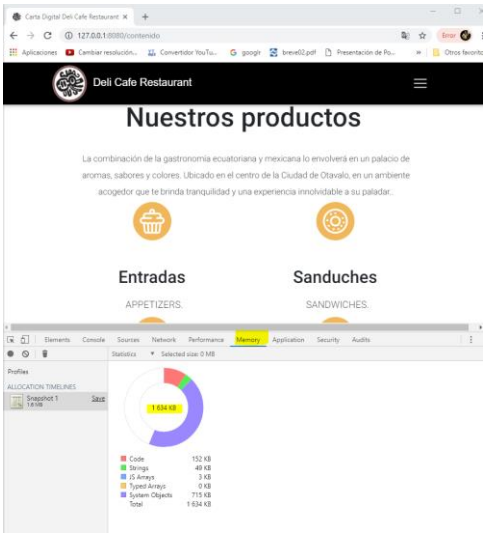



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**156.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 325ms como se muestra en la Figura 33:

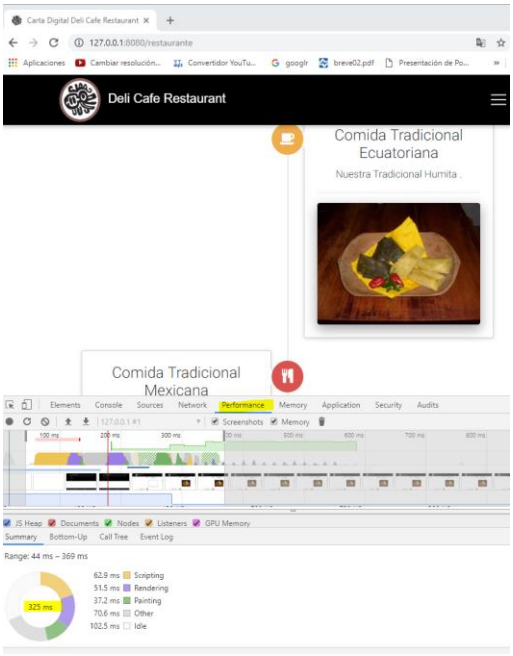



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 19:**

**157.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 699ms como se muestra en la Figura 34:

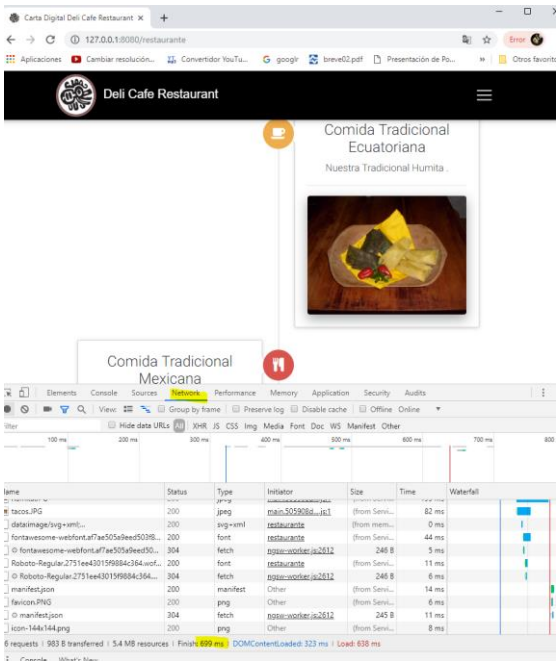



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**158.** **Memoria:** Al hacer la siguiente iteración consume1634 kb como se muestra en la Figura 32.

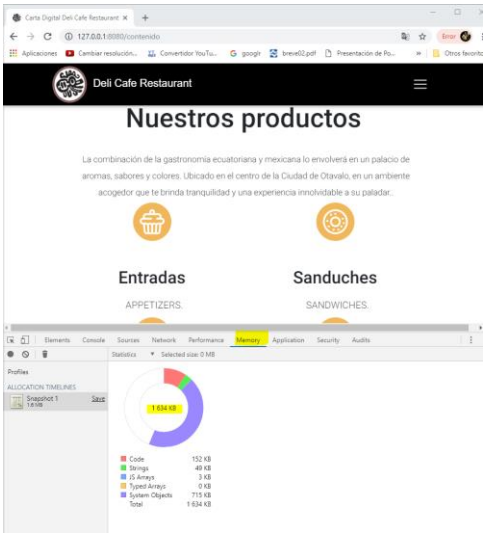



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**159.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 325ms como se muestra en la Figura 33:

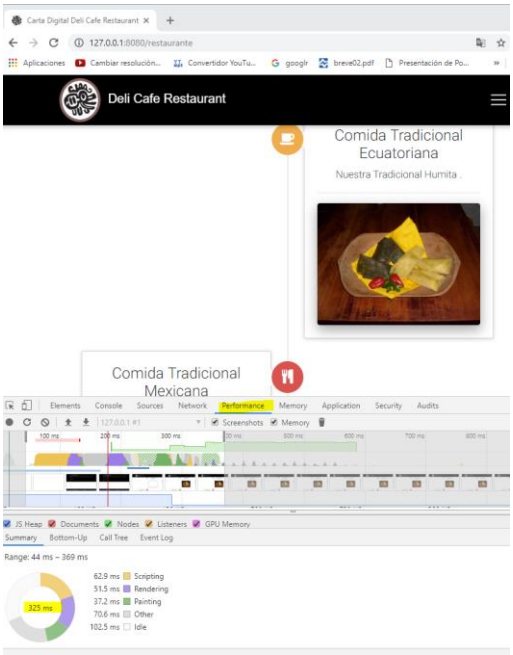



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

- **Iteración nro. 20:**

**160.** **Red:** Al hacerla la siguiente iteración se realiza en un tiempo 694ms como se muestra en la Figura 34:

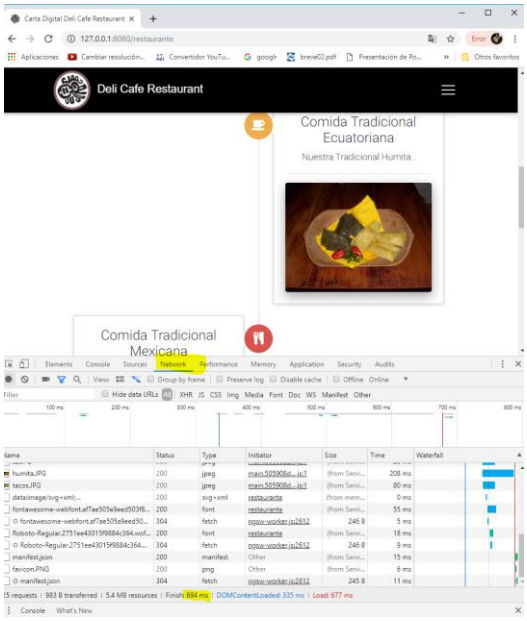



**Fig. 37.** Arquitectura App Shell-Red

**Fuente:** Propia

**161.** **Memoria:** Al hacer la siguiente iteración consume1634 kb como se muestra en la Figura 32.

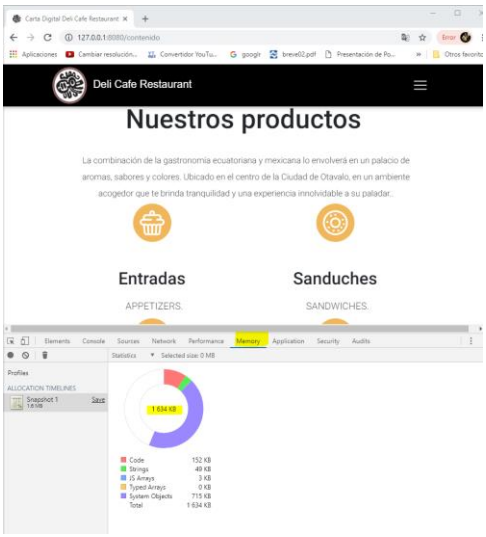



**Fig. 38.** Arquitectura App Shell-Memoria

**Fuente:** Propia

**162.** **Rendimiento:** Al hacerla la siguiente iteración se realiza en un tiempo 325ms como se muestra en la Figura 33:

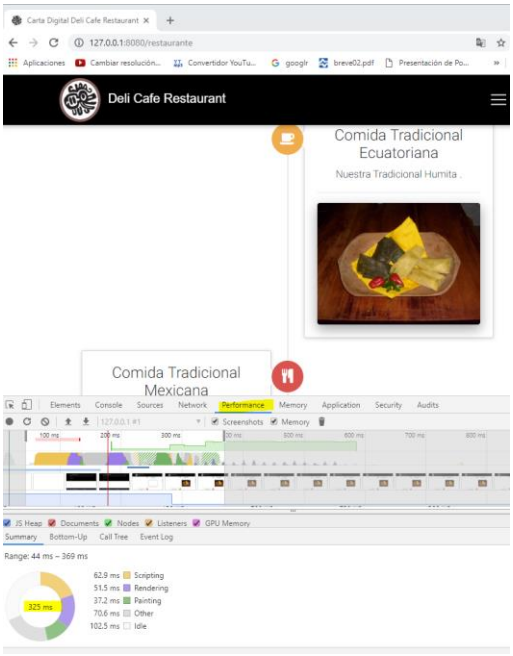



**Fig. 39.** Arquitectura App Shell-rendimiento

**Fuente:** Propia

**Anexo 3: Encuesta**

Estudio de las Arquitecturas Tecnológicos (App Shell, Responsive Design, SOA) en la construcción de Web Progresiva para fortalecer la gestión de pedidos de los locales Mipymes de tipo Café-Restaurant en la ciudad de Otavalo.

Importante: Una aplicación Web Progresiva utiliza las últimas tecnologías disponibles en los navegadores cuya finalidad es ofrecer una experiencia en móviles o cualquier otro dispositivo sin necesidad de una conexión a internet.

Instructivo para llenar la encuesta:

- Lea las preguntas atentamente, revise todas las opciones y elija la respuesta que prefiera en las preguntas de selección.
- Piense antes de contestar
- La encuesta será asistida mediante correo electrónico.

Rol, Cargo u Ocupación:

_____

Fecha:

_____

11. ¿Se ha presentado problemas relacionados al proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?

    Si: No:

12. ¿Ha existido inconvenientes en la escasez de menús o cartas físicas cuando el restaurante se encuentra lleno en su totalidad, dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?

    Si: No:

13. ¿Ha existido inconvenientes en la toma de un pedido a los clientes dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?

    Si: No:

14. ¿Se ha presentado inconsistencias y retrasos en la entrega del pedido dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?

    Si: No:

15. ¿Ha existido problemas en la publicación del menú dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?

    Si: No:

16. ¿Cree que sería necesario aplicar herramientas tecnológicas en el proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant

    Si: No:

17. ¿Cree que sería necesaria la implementación de una aplicación Web y que a su vez sea Progresiva para optimizar el proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?

    Si: No:

18. ¿Cree que sería ventajoso que la aplicación Web Progresiva funcione dentro y fuera del establecimiento con o sin conexión a internet dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?

    Si: No:

19. ¿Indique los beneficios más relevantes que tendría la implementación de una aplicación Web Progresiva dentro del proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?

    - Reduce gastos de impresión y de papel.
    - La posibilidad de tener organizada y actualizada la información más rápidamente.

- Facilita la elección de una forma atractiva, sencilla, intuitiva, con una información más completa que proporciona al cliente un mayor grado de acierto y satisfacción con su selección.

20. ¿Estaría dispuesto a utilizar una aplicación Web Progresiva que permita optimizar el proceso de gestión de pedidos de los locales MiPymes de tipo Café-Restaurant?

Si: No:

**Anexo 4: Evidencia de la investigación**