



FACULTAD 1

# Aplicación para la detección de redirecciones en el Directorio de Sitios Web

Trabajo de diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autor(es):** Iris Leybis Atencio Ramírez

**Tutor(es):**

MSc. Yalice Gámez Batista

Ing. Xavier Soto Rizo

La Habana, diciembre de 2022

Año 64 de la Revolución.

## **DECLARACIÓN DE AUTORÍA**

Se declara que Iris Leybis Atencio Ramírez es la única autora del presente trabajo de Diploma que tiene por título “Aplicación para la detección de redirecciones en el Directorio de Sitios Web” y se concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, para hacer uso en su beneficio.

Para que así conste firmamos la presente a los 2 días del mes de diciembre del año 2022.

Iris Leybis Atencio Ramírez

---

Firma del Autor

MSc. Yalice Gámez Batista

---

Firma del Tutor

Ing. Xavier Soto Rizo

---

Firma del Tutor

## **DEDICATORIA**

Este trabajo de diploma está dedicado a las personas más importantes de mi vida:

A mi mamá, mi papá y mi hermana por su apoyo incondicional en todas las decisiones que he tomado.

A mis abuelos, principalmente a Nancy por siempre estar ahí. A mi compañero de vida Eddy por no dejarme rendir nunca y luchar cada segundo conmigo en mis altas y bajas de la carrera.

## **AGRADECIMIENTOS**

Agradecida especialmente con mis padres por el amor y la educación que me han dado e incitarme siempre a ser una profesional.

A mi hermana por estar siempre ahí, con su apoyo incondicional.

A mis abuelos y tíos por su cariño, apoyo y consejos llenos de sabiduría.

A mis primos por permitirme ser un ejemplo para ellos.

A mis suegros por acogerme como una hija y por la confianza.

Agradecida con mis tutores Xavier y Yalice por darme la oportunidad y por saber guiarme durante todo este proceso.

A los profes José Gabriel y Rubén por su incondicional apoyo durante este proceso.

A mis compañeros de aula especialmente a Liuba que más que una compañera es una hermana para mí, gracias por tu ayuda y apoyo durante estos 5 años de carrera.

A mis amigos que son como familia Lázara, Madelín, Dariel, Celeste, Laura, Julio, gracias le doy a la vida por haberlos conocido.

A todos los profesores que formaron parte de mi formación profesional por sus enseñanzas y consejos siempre para ser mejor.

Y por último pero no menos importante a mi novio Eddry por ser mi apoyo en todo momento y no dejarme rendir, por tantas noches de desvelo, por su ayuda, compañía y por saber comprenderme.

## **RESUMEN**

Con el propósito de aprovechar al máximo las TIC, y para mostrar a los usuarios la información existente en el dominio cubano que Google ignora o invisibiliza, nace el proyecto Monitor Red Cuba. Este tiene como objetivo de aumentar la visibilidad y el alcance de los contenidos cubanos mediante la activación de técnicas de posicionamiento web. Dentro de este proyecto se encuentra la aplicación directorio de sitios web cubanos que permite la caracterización de la red nacional, así como clasificar y gestionar las direcciones web que están indexadas en el Monitor. Dentro de los indicadores que se identifican en los sitios web se encuentra el tipo de sitio, el cual puede ser clasificado en dos grupos: Portales o Aplicaciones, y a su vez pueden tener asociadas direcciones web que son redirecciones o réplicas de estos portales o aplicaciones.

La presente investigación tiene como objetivo desarrollar una aplicación que permita detectar las redirecciones en los sitios web indexados en el monitor para lograr un mejor posicionamiento de los mismos. Para el desarrollo de este sistema se utilizó como metodología de desarrollo XP o Programación Extrema, como lenguaje de programación: Java 11.0 y de modelado UML. Se usa como plataforma y framework de desarrollo: Spring boot 1.5.8, como entorno de desarrollo NetBeans 12.1, como herramienta CASE el Visual Paradigm 8.0, para el control de versiones Git (Gitlab de la Universidad), y como gestor de base de datos MySQL.

**Palabras clave:** directorio, microservicios, posicionamiento web, redirecciones.

## **ABSTRACT**

With the purpose of making the most of ICT, and to show users the existing information in the Cuban domain that Google ignores or makes invisible, the Monitor Red Cuba project was born. This aims to increase the visibility and reach of Cuban content by activating web positioning techniques. Within this project is the application directory of Cuban websites that allows the characterization of the national network, as well as classifying and managing the web addresses that are indexed in the Monitor. Among the indicators that are identified in the websites is the type of site, which can be classified into two groups: Portals or Applications, and in turn may have associated web addresses that are redirects or replicas of these portals or applications.

The objective of this research is to develop an application that allows the detection of redirections in the websites indexed in the monitor to achieve a better positioning of the same. For the development of this system, XP or Extreme Programming was used as development methodology, as programming language: Java 11.0 and UML modeling. It is used as development platform and framework: Spring boot 1.5.8, as NetBeans 12.1 development environment, as CASE tool Visual Paradigm 8.0, for Git version control (Gitlab of the University), and as database manager MySQL.

**Keywords:** directory, microservices, redirections, web positioning.

**ÍNDICE**

INTRODUCCIÓN .....	10
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA .....	15
I.1 Conceptos asociados a la investigación .....	15
I.2 Análisis de las soluciones existentes .....	20
I.2.2 Resultado del estudio de Sistemas homólogos .....	23
I.3 Entorno de desarrollo de la propuesta de solución .....	23
I.3.1 Metodología de desarrollo de software .....	24
I.3.2 Lenguajes.....	25
I.3.3 Herramientas y tecnologías .....	26
Conclusiones del capítulo.....	28
CAPÍTULO II: PLANIFICACION Y DISEÑO DE LA SOLUCIÓN .....	29
II.1 Propuesta de solución .....	29
II.1.1 Modelo conceptual .....	29
II.1.2 Requisitos funcionales.....	30
II.1.3 Requisitos no funcionales.....	31
II.1.3 Historias de usuarios .....	31
II.1.4 Plan de iteraciones .....	35
II.2 Diseño .....	36
II.2.1 Arquitectura de software.....	36
II.2.2 Diagrama de clases.....	37
II.2.3 Patrones de diseño.....	38
II.2.3.1 Patrones de diseño GRASP .....	38
II.2.3.1 Patrones de diseño GOF .....	40
II.2.3 Tarjetas Clase-Responsabilidad-Colaborador .....	40
II.3 Modelo de datos.....	42
Conclusiones del capítulo.....	43
CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA .....	44
III.1 Diagrama de componentes.....	44
III.2 Estándares de codificación.....	45
III.3 Pruebas de software.....	46

III.3.1 Pruebas unitarias.....	47
III.3.2 Pruebas de integración.....	50
III.3.3 Pruebas de aceptación.....	51
III.4 Diagrama de despliegue.....	53
Conclusiones del capítulo.....	53
CONCLUSIONES GENERALES.....	54
RECOMENDACIONES.....	55
REFERENCIAS BIBLIOGRÁFICAS.....	56

## ÍNDICE DE TABLAS

Tabla 1: Relación de medidores.....	20
Tabla 2: Relación de medidores y homólogos.....	23
Tabla 3: HU1.....	32
Tabla 4: HU2.....	33
Tabla 5: HU3.....	33
Tabla 6: HU4.....	34
Tabla 7: HU5.....	34
Tabla 8: HU6.....	35
Tabla 9: Plan de iteraciones.....	35
Tabla 10: Tarjeta CRC1 .....	41
Tabla 11: Tarjeta CRC2 .....	41
Tabla 12: Tarjeta CRC3 .....	41
Tabla 13: Tarjeta CRC4 .....	42
Tabla 14: Tarjeta CRC5 .....	42
Table 15: Tarjeta CRC6 .....	42
Tabla 16: Cálculo de la complejidad ciclomática del fragmento del método getUrlConection. ....	47
Tabla 17: Caminos del grafo de flujo.....	48
Tabla 18: Caso de prueba Protocolo redirigido. EC1.1.....	48
Tabla 19: Caso de prueba Protocolo redirigido. EC1.2.....	49
Tabla 20: Caso de prueba Protocolo redirigido. EC1.3.....	49
Tabla 21 Caso de prueba integración de la solución.....	50

## ÍNDICE DE FIGURAS

Figura 1:Partes de un proceso. Fuente: (Llamas, 2021).....	16
Figura 2: Modelo conceptual. Fuente: (Elaboración propia) .....	30
Figura 3 Arquitectura de microservicios. Fuente (Diego Navarro y Cabrera Lozada, 2020) .....	37
Figura 4: Diagrama de clases. Fuente: (Elaboración propia).....	38
Figura 5 Modelo de datos. Fuente: (Elaboración propia).....	43
Figura 6 Diagrama de componentes. Fuente: (Elaboración propia) .....	44
Figura 7 Resultados de las pruebas de integración. Fuente (Elaboración propia) .....	51
Figura 8 Aval de aceptación de la solución. ....	52
Figura 9 Diagrama de despliegue. Fuente: (Elaboración propia).....	53

## INTRODUCCIÓN

En la actualidad se ha evidenciado un acelerado desarrollo de las tecnologías de la información y las comunicaciones (TIC), que ha traído consigo un aumento del uso y desarrollo de portales y sitios web. Esto ha generado la necesidad de herramientas de búsqueda de información y sitios web que permitan a los usuarios un mejor acceso a la información. En nuestro país se ha ido desarrollando una revolución tecnológica por parte de las empresas, entidades y ciudadanos donde el uso de la web ha contribuido a una mayor interacción con el mundo de las redes digitales, así como una posibilidad de llegar a una mayor cantidad de personas y posibles clientes o potenciales inversionistas para la divulgación de sus productos y servicios.

Con el propósito de aprovechar al máximo las TIC, y para mostrar a los usuarios la información existente en el dominio cubano que Google ignora o invisibiliza, nace el proyecto Monitor Red Cuba. Este tiene como objetivo aumentar la visibilidad y el alcance de los contenidos cubanos mediante la activación de técnicas de posicionamiento web, que influyen directamente en el incremento de la relevancia y popularidad de los sitios web. Con este fin, se provee a los diversos usuarios de un grupo de saberes, aplicaciones y herramientas que les permiten a los sitios web elevar su calidad y con ello, la visibilidad de sus contenidos. Fue desarrollado por la Universidad de las Ciencias Informáticas desde el año 2017 en la dirección de Proyectos Especiales. Este macroproyecto está conformado por cuatro aplicaciones principales que permiten brindar el servicio de posicionamiento web o SEO (como es conocido según las siglas en inglés de Search Engine Optimization) (Guerra Soriano, 2021).

1. **Evaluador de sitios web:** realiza un diagnóstico continuo a los sitios indexados en las aplicaciones para analizar cuestiones asociadas al nivel de optimización e implementación de técnicas de posicionamiento y de la infraestructura que los soporta.
2. **SEO WebMas:** es una aplicación de análisis en línea que muestra el resultado de la evaluación realizada a cada sitio web indexado y tributa un grupo de buenas prácticas a considerar por los usuarios.
3. **Telus:** herramienta cuyo objetivo es analizar el tráfico generado hacia el sitio web, estudiar los públicos, así como los contenidos y servicios más consumidos por los usuarios.

**4. Directorio de sitios web:** permite la caracterización de la red nacional.

La presente investigación se centra en el Directorio de Sitios Web, el cual representa la forma de entrada de los sitios que serán evaluados por el resto de las aplicaciones.

Mediante la aplicación Directorio de Sitios Web, el grupo de seguimiento de la Dirección de Proyectos Especiales clasifica y gestiona las direcciones web que están indexadas en el Monitor de Sitios Web, y configura los nomencladores a utilizar en la clasificación de las direcciones web.

Dentro de los indicadores que se identifican en los sitios web se encuentran: el nombre del sitio, la provincia, organismos e institución a la que pertenece, la temática que aborda y el tipo de sitio, el cual puede ser clasificado en dos grandes grupos: Portales o Aplicaciones y a su vez pueden tener asociadas direcciones web que son redirecciones o réplicas de estos portales o aplicaciones.

Actualmente, una vez que la dirección web es detectada por los rastreadores, los especialistas deben revisarla manualmente para clasificarla. En caso de que la dirección encontrada sea una redirección a otra dirección ya registrada anteriormente, sin que se haya establecido una relación entre ambas, el especialista debe buscar el sitio original y dejar explícito en el nombre que se trata de una redirección, pero si ello variara sería imperceptible, pudiendo provocar errores en la clasificación dada lo que hace el proceso de trabajo más complejo y agotador pues se realiza de forma manual.

Para identificar si un sitio posee una redirección, el especialista debe consultar el código de estado de la URL y si es 200 no posee redirección, pero si es 301, 302, 303, 307 o 308 es que el sitio redirecciona a otro. Además, se debe realizar el mismo proceso para la URL a la que redirecciona el sitio original hasta llegar a la última URL donde el código de estado va a ser 200. Al día puede que estarse analizando muchos sitios por lo que puede ser muy difícil clasificar una gran cantidad. Una clasificación errónea como consecuencia afecta el posicionamiento del sitio.

Partiendo de esta situación se identifica como **Problema de investigación:** ¿Cómo contribuir a la mejora del proceso de detección de redirecciones en el Directorio de Sitios web perteneciente al monitor Red CUBA?

Siendo el **Objeto de estudio** de esta investigación: Los procesos para la detección de redirecciones en sitios web.

Cuyo **Campo de acción** está enmarcado en las aplicaciones para la detección de redirecciones en el Directorio de Sitios Web perteneciente al Monitor Red CUBA.

Para dar solución a la problemática planteada la presente investigación tiene como **Objetivo general**: desarrollar una aplicación que permita la detección de redirecciones en el Directorio de Sitios Web perteneciente al Monitor Red CUBA.

Las **preguntas científicas** que guían y orientan el desarrollo del proceso investigativo son las siguientes:

- ¿Cuáles son los fundamentos teórico-metodológicos a tener en cuenta para la construcción del marco teórico-conceptual a utilizar en la presente investigación?
- ¿Cuál es el estado inicial del proceso para la detección de redirecciones en el Directorio de Sitios Web perteneciente al Monitor Red CUBA?
- ¿Cómo implementar una aplicación para la detección de redirecciones en el Directorio de Sitios Web perteneciente al Monitor Red CUBA?
- ¿Cuál es la validez de la aplicación para la detección de redirecciones en el Directorio de Sitios Web perteneciente al Monitor Red CUBA?

Para darle solución a los objetivos planteados se proponen las siguientes **Tareas de Investigación**:

1. Determinación de los fundamentos teórico-metodológicos a tener en cuenta para la construcción del marco teórico-conceptual a utilizar en la presente investigación.
2. Determinación del estado inicial del proceso para la detección de redirecciones en el Directorio de Sitios Web perteneciente al Monitor Red CUBA.
3. Implementación de una aplicación para la detección de redirecciones en el Directorio de Sitios Web perteneciente al Monitor Red CUBA.
4. Validación de la aplicación para la detección de redirecciones en el Directorio de Sitios Web perteneciente al Monitor Red CUBA.

Para el desarrollo de la investigación se hizo uso de métodos científicos de investigación siendo estos la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. Se clasifica en teóricos y empíricos, los cuales están dialécticamente relacionados (Hernández León y Coello González, 2012). En la investigación se utilizaron los siguientes **métodos teóricos**.

**Analítico-Sintético**: Se utilizó para la recopilación de información requerida durante la realización del estudio del arte y para el desarrollo del trabajo mediante la revisión de

documentos y artículos de donde se extrajeron los elementos más significativos relacionados con los sistemas de detección de redirecciones, además del análisis de las diferentes metodologías, tecnologías y herramientas a utilizar para el desarrollo de la misma (Raúl, 2009).

**Histórico-lógico:** Se empleó en el estudio de cómo ha evolucionado el tema que se está investigando, las diferentes herramientas existentes y sus características, lo que ayuda a la comprensión de la necesidad del sistema.

**Modelado:** Se utilizó para modelar todos los diagramas correspondientes a la etapa de análisis, diseño e implementación del sistema a desarrollar; las relaciones entre objetos y las actividades que intervinieron en el proceso de configuración del entorno colaborativo en la presente investigación.

Los **métodos empíricos** usados fueron:

**Entrevista:** Se realizó entrevistas a los trabajadores del Monitor de Sitios Web Cubano para constatar las condiciones en las que se encuentra Directorio de Sitios web y la importancia que tiene actualmente la aplicación de detección de redirecciones.

**Observación:** Se utilizó para el diagnóstico del problema que se investiga, a fin de obtener los elementos necesarios para desarrollar la aplicación de detección de redirecciones. Se verá como resultado de esta observación la necesidad del desarrollo de una aplicación para la detección de redirecciones al Directorio de Sitios web.

### **Estructura de capítulos:**

El documento está estructurado por tres capítulos, las conclusiones generales, las referencias bibliográficas y los anexos. La estructura de los capítulos se define a continuación:

#### **Capítulo 1: Fundamentación Teórica**

En este capítulo se abordan los principales presupuestos teóricos asociados a la investigación, se realiza un estudio de algunos sistemas que poseen un comportamiento similar al proceso a desarrollar, y además se describe la metodología de desarrollo a implementar, las herramientas informáticas y los lenguajes a utilizar para el desarrollo de la aplicación.

## **Capítulo 2: Planificación y Diseño de la solución**

Se realiza una descripción detallada de la propuesta de solución definiendo los elementos técnicos de la misma: los patrones, el diseño de clases, los medios empleados y las restricciones de diseño. Se representan mediante un modelo de dominio los principales conceptos que se manejan en el contexto del sistema. Se realiza una descripción de los requisitos funcionales y se especificaron en las Historias de Usuario como dicta la metodología.

## **Capítulo 3: Validación de la solución propuesta**

Este capítulo comprende todo lo relacionado con las pruebas efectuadas a la implementación, para poder medir la calidad y eficiencia de acuerdo con las necesidades que requiere el cliente. Se representan los elementos físicos necesarios para un correcto despliegue de la aplicación, empleando para ello el diagrama de despliegue. Se muestran componentes de la implementación. Se realiza la validación y prueba de la solución de acuerdo a los requisitos que debe cumplir para garantizar una calidad óptima.

## **CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA**

En el presente capítulo se detallan los elementos teóricos que apoyan la investigación y el desarrollo del tema profundizándose en los aspectos y conceptos más importantes para la investigación. Además, se realiza un análisis sobre la metodología de desarrollo de software utilizada, así como las herramientas y tecnologías necesarias para llevar a cabo el desarrollo e implementación del proceso de detección de redirecciones en el Directorio de Sitios Web.

### **I.1 Conceptos asociados a la investigación**

En el presente epígrafe se analizan los conceptos fundamentales asociados a la investigación para una mejor comprensión de la misma.

#### **Proceso**

Un proceso informático consiste en la realización de una serie de tareas e instrucciones que conllevan a un resultado final.

A nivel informático, para que el proceso se dé debe realizarse una entrada de datos, y, mediante un programa informático, tratar esos datos vía instrucciones y cálculos del programa. Esto, para luego revelar una salida de datos final, la cual será el resultado de todo el proceso.

El proceso informático puede darse ,bien porque el usuario ejecute un programa, o bien porque el ordenador o computadora arranque por sí mismo un programa de forma automática. Luego, la entrada de datos no siempre tiene que ser manual por parte del usuario (Llamas, 2021).

A nivel esquemático, se puede mostrar de forma general cómo funciona un proceso informático de la siguiente forma (ver Figura 1).



**Figura 1:Partes de un proceso. Fuente: (Llamas, 2021)**

De acuerdo con lo planteado anteriormente se puede concluir que un proceso, asociado a la presente investigación es la secuencia de acciones que permiten que se realice la detección sitios que poseen redirecciones en la aplicación general Directorio de Sitios Web.

### **Directorios**

Los directorios son listas de recursos organizados por categorías temáticas que se estructuran jerárquicamente en un árbol de materias que permite visualizar los recursos descendiendo desde los temas más generales situados en las ramas superiores, a los temas más específicos situados en las ramas inferiores. Las categorías ofrecen una lista de enlaces a las páginas que aparecen referenciadas en el buscador. Cada enlace también ofrece una breve descripción de su contenido. Así pues, los directorios o índices se estructuran por temas o categorías principales que, a su vez, contienen otras subcategorías, y así sucesivamente hasta que al final se ofrecen enlaces directos a otras páginas o recursos de Internet (Lapiente, 2013).

Es importante conocer el concepto de directorio pues precisamente la solución propuesta resuelve un problema para la aplicación principal Directorio de Sitios Web cubanos.

### **Redirección**

Una redirección es el proceso de reenviar un cliente desde la URL solicitada a otra URL. Las redirecciones se utilizan normalmente porque el documento se ha movido temporal o permanentemente a otra URL. Los redireccionamientos pueden ser una herramienta eficaz para garantizar la usabilidad y el SEO.

Una redirección es un reenvío automático desde la URL solicitada a una URL de destino. La redirección se completa cuando el cliente se reenvía a la URL de destino de la redirección (Schwarz y Müller, 2022).

Existen redirecciones del lado del servidor, con esta forma se introduce un comando correspondiente en el archivo htaccess del servidor. Si se llama a la URL el servidor redirigirá inmediatamente al nuevo destino.

Este tipo de redirecciones son:

- Redirección 301: este código indica una redirección permanente del dominio. Al ingresar a una página con el código 301 activo, inmediatamente el usuario será redireccionado a otro.
- Redirección 302: este código es una redirección temporal muy similar a la anterior.
- Redirección 307: esta redirección se da cuando petición fue procesada con una URL antigua pero que aún está en funcionamiento y además cuenta con una URL nueva.

Las redirecciones del lado del cliente son otra forma de redirecciones donde no es necesario realizar ningún cambio en el servidor, ya que los redireccionamientos se almacenan directamente en el código fuente de la página.

Un ejemplo de este tipo de redirección es Meta Refresh que se configura colocando una metaetiqueta<sup>1</sup> en el encabezado del documento HTML.

Es importante conocer claramente lo que es una redirección porque es el objeto fundamental de esta solución. Esta aplicación va a tener la capacidad de reconocer todos los tipos de redirecciones que pueden tener los sitios web, ya sean del lado del servidor o del lado del cliente.

## **Sitio web**

Los sitios Web son estructuras de información como tantas otras, con las singularidades que aportan las características que le son propias como: la hipertextualidad, la forma gráfica, el acceso y la interactividad. En consecuencia, al igual que cualquier otro documento deben ser pensados críticamente y concebidos en términos de proyecto (García de León, 2002).

---

<sup>1</sup> Las metaetiquetas son etiquetas HTML que se incorporan en el encabezado de una página web y que resultan invisibles para un visitante, pero de gran utilidad para navegadores u otros programas que puedan valerse de esta información (Barzanallana, 2017)

## Posicionamiento web

Posicionamiento web, posicionamiento en buscadores o posicionamiento SEO se refiere a las técnicas para que una página web aparezca en las primeras posiciones de los resultados en buscadores (como Google) para una serie de palabras o frases (Urosa Barreto, 2021). El SEO trabaja en aspectos técnicos como la optimización de la estructura de una web, pero también se aplica a nivel de contenidos, con el objetivo de volverlos más útiles y relevantes para los usuarios.

El posicionamiento natural u orgánico es la que consigue una web de forma espontánea usando técnicas de posicionamiento web que tienen que ver con los contenidos de la misma (títulos, metas, palabras clave, etcétera), con los enlaces dirigidos a la página web y con la calidad del código, sin necesidad de que medie una campaña de publicidad o campañas de posicionamiento de pago por clic que paga directamente a los buscadores para aparecer como un enlace patrocinado: Google Adwords, Yahoo! Search Marketing, entre otras (Mousinho, 2020).

El SEO orgánico o libre de pago consiste en la práctica activa de la optimización de un sitio web mediante la mejora de aspectos internos y externos con el fin de aumentar el tráfico que una página web recibe desde los motores de búsqueda sin necesidad de pagarles a los mismos.

Características propias del SEO:

- Palabras clave: Saber y comprender como las personas están buscando contenidos.
- Estructura de la página: Actualmente, los motores de búsqueda igualmente otorgan mucha desazón a la red del punto, ya que es esta la que determina qué tan admisiblemente está estructurada.
- Meta Descripciones: La posición es como un lugar que ayuda a los usuarios a identificar el contenido de tu emplazamiento desde que están revisando el listado de su investigación.
- Calidad Del Contenido: El objetivo específico debe ser que el contenido ayude al usuario sin importar si se trata de texto, video o una figura.

Ventajas del uso de SEO:

- Posicionamiento a largo plazo: Los sitios se mantienen más tiempo en las primeras páginas de los buscadores.
- Aumento del tráfico objetivo: Los sitios obtendrán más visitas de usuarios.

- Aumentar ventas o conversiones: Al tener más visitas los negocios virtuales tendrán más ventas.
- Creación de buenos contenidos: El sitio web tendrá mejor calidad y prestigio.

El SEO es una forma de verificar que tan bien hecho y optimizado está un sitio web, lo cual ayuda a ganar usuarios y visitas a los sitios con el uso de las tecnologías móviles y el crecimiento del comercio electrónico.

## Endpoint

Cuando se habla de endpoint o punto final se hace referencia a un extremo de un canal de comunicación. Cuando una API (Application Programming Interfaces, que en español significa interfaz de programación de aplicaciones) interactúa con otro sistema, los puntos de contacto de esta comunicación se consideran puntos finales. Para las API, un punto final puede incluir una URL de un servidor o servicio. Cada punto final es la ubicación desde la cual las API pueden acceder a los recursos que necesitan para llevar a cabo su función.

Las API funcionan mediante "solicitudes" y "respuestas". Cuando una API solicita información de una aplicación web o un servidor web, recibirá una respuesta. El lugar donde las API envían solicitudes y donde vive el recurso se denomina punto final (*API Endpoints*, 2022).

Para integrar la solución propuesta con el directorio de sitios web cubanos es necesario utilizar una interfaz de programación de aplicaciones (API). En este caso será integrado con un endpoint de API que es el lugar donde se realizan esas solicitudes (conocidas como llamadas API).

## URL

URL es una sigla del idioma inglés correspondiente a **Uniform Resource Locator (Localizador Uniforme de Recursos)**. Se trata de la secuencia de caracteres que sigue un estándar y que permite denominar recursos dentro del entorno de Internet para que puedan ser localizados.

El URL es, por lo tanto, el conjunto de caracteres que posibilita la asignación de una dirección exclusiva a un recurso que se encuentra disponible en el espacio virtual. En otras palabras, el URL es una dirección de Internet que, al ser encontrada y visualizada por un navegador, muestra un recurso de información al usuario (Pérez Porto, 2010).

Es fundamental conocer el significado de URL pues en la aplicación para la detección de redirecciones este será el elemento que tomaremos de los sitios web para realizar las

consultas al código de estado, de esta forma conoceremos si el sitio posee redirección y en caso de la tenga a que URL nos lleva dicha redirección.

## I.2 Análisis de las soluciones existentes

### Sistemas de homólogos

Una de las primeras etapas que debe desarrollarse dentro de una investigación es el estudio del estado del arte ya que permite determinar la forma como ha sido tratado el tema, cómo se encuentra el avance del conocimiento en el momento de realizar una investigación y cuáles son las tendencias existentes para el desarrollo de la temática o problemática que se va a llevar a cabo. El estado del arte sirve al investigador como referencia para asumir una postura crítica frente a lo que se ha hecho y lo que falta por hacer en torno a una temática o problemática concreta, para evitar duplicar esfuerzos o repetir lo que ya se ha dicho, y además para localizar errores que ya fueron superados.

A partir de los intercambios realizados con el cliente y los especialistas del proyecto se definieron los indicadores que posibilitaron analizar la viabilidad de usar alguna de estas aplicaciones. Estos indicadores se describen en la Tabla número 1.

Tabla 1: Relación de medidores.

No	Medidores
1	Capacidad de acoplarse al sistema.
2	Compatibilidad con los lenguajes y herramientas a usarse.
3	Satisfacción de los requisitos de obtención de la información que se necesita.

La capacidad de acoplarse al sistema es la forma y nivel de interdependencia entre módulos de software. Es una medida de qué tan cercanamente conectados están uno o más módulos de software.

La compatibilidad con los lenguajes y herramientas a usarse se define como medidor porque la solución va a estar integrada a un sistema y por tanto las herramientas y lenguajes que se usen deben ser compatibles con este.

Los requisitos de obtención de la información que debe cumplir un software para que cumpla con las necesidades de la problemática planteada es que dada una URL de un sitio web saber si posee una redirección, en caso de que la tenga especificar a qué sitio redirecciona y volver a consultar hasta llegar a la URL final del sitio.

Al realizar el estudio del estado del arte se identificaron un conjunto de aplicaciones que pueden ser utilizadas como alternativas para la detección de redirecciones se encontraron varios softwares y herramientas que detectan cuando un sitio posee redirección.

### **Geekflare Tools**

Conjunto de herramientas gratuitas para probar su sitio en busca de problemas de DNS, seguridad, rendimiento, red y SEO.

En la sección de SEO se puede encontrar una herramienta para comprobar redirecciones de sitios web. Esta herramienta está diseñada para rastrear la ruta de la URL con el código de estado y los encabezados. En pocas palabras, muestra todos los saltos necesarios para que la URL finalmente se cargue en el navegador desde el momento en que la ingresa en la barra de direcciones. Esto puede ayudar a solucionar los problemas de los enlaces redirigidos para ver si funcionan según lo previsto y no van por el camino 404. Además, también puede usar esto para verificar la ruta de URL acortadas. Esto puede ser útil para ver la ruta completa y el servidor final al que apunta una URL corta, puede verificar el código de estado de las redirecciones y optar por el servicio de acortamiento de URL utilizando redirecciones 301 para obtener la máxima ventaja de SEO. Además de esto, uno puede usar el Comprobador de redirección de sitios web para rastrear las redirecciones para eliminar los rastreadores y usar la fuente web prevista sin ningún tipo de registro de datos (URL Redirection Checker, s.f.).

El estudio de esta herramienta aportó un mejor entendimiento sobre las redirecciones y de cómo funcionan las herramientas destinadas a comprobar redirecciones de URL. No es recomendable el uso de esta herramienta en el directorio de sitios web porque analiza una URL a la vez y no devuelve el destino final de la misma.

### **Site24x7**

Es un servicio de monitoreo de aplicaciones, redes, servidores y recursos web a través de Internet. Con un modo de pago por uso y un abanico de prestaciones avanzadas para la monitoreo y análisis de recursos web, se trata de un componente imprescindible dentro de cualquier estrategia de gestión de infraestructuras.

Site24x7 aporta una nueva dimensión a sus servicios de monitoreo, ya que permite analizar la experiencia real de usuarios a través de Internet desde más de 40 ubicaciones en todo el mundo. Módulos avanzados como APM (Application Performance Management) permiten incorporar elementos estratégicos como la medición de la calidad de servicio, el

análisis de rendimiento en las transacciones específicas y otras ventajas(Website Monitoring Service: Site24x7, s.f.).

Este posee un grupo de herramientas gratuitas organizadas por categorías donde se pueden encontrar herramientas de validación y dentro de estas un verificador de redirecciones, donde se incerta la URL del sitio, la ubicación de la prueba y se configura un agente de usuario del cliente para simular la aplicación de su cliente o un navegador.

Esta herramienta es muy útil para detectar redirecciones simulando aplicaciones, por ello no es posible su uso en el directorio y sería muy trabajoso realizar grandes cambios para adaptarlo a la problemática identificada.

### **Wherergoes**

Este comprobador de redirecciones de URL mostrará la ruta de redirección completa de URL, enlaces acortados o URL diminutas.

Cuando el verificador de enlaces rastrea un enlace, los resultados muestran todos los enlaces a través de los cuales está siendo redirigido, incluidos los redireccionamientos de php, los redireccionamientos de htaccess, los redireccionamientos de NGINX, los redireccionamientos de JavaScript y las meta actualizaciones que eventualmente lo llevarán al destino final.

Esta herramienta de rastreo de redireccionamientos le permite solucionar problemas de enlaces y realizar inteligencia competitiva de una manera que antes no era posible (*Redirect Checker - WhereGoes, 2022*).

El estudio de esta herramienta brindó una mejor comprensión sobre el proceso que se desea desarrollar en el directorio de sitios web, pero su uso sería imposible ya que hay introducir de forma manual la URL que se quiere analizar.

### **Redirect-checker**

Este verificador de redirecciones tiene la opción de insertar la URL del sitio y como opciones del verificador debe comprobar su redireccionamiento con un agente de usuario específico, seleccionando el agente de usuario del navegador para probar su redirección (*Redirect Checker, s.f.*).

Al igual que en el verificador anterior que analiza una URL a la vez y esta tiene que ser introducida por el usuario. Esta herramienta solo comprueba redirecciones de tipo 301 y 302 por lo que no sirve como solución pues es necesario detectar todo tipo de redirecciones.

## I.2.2 Resultado del estudio de Sistemas homólogos

Una vez analizadas las aplicaciones identificadas, se concluye que ninguna puede ser utilizada completamente como solución, pues no cumplen con todos los requisitos necesarios para realizar un proceso de detección de redirecciones para el Directorio de Sitios Webs, por lo que se determina la necesidad de desarrollar una aplicación para la detección de redirecciones en el directorio de sitios web cubano. En la Tabla que se muestra a continuación (Tabla 2) se muestra un resumen de la evaluación de los sistemas para la detección o verificación de redirecciones analizados.

**Tabla 2: Relación de medidores y homólogos.**

Homólogo	Medidor 1	Medidor 2	Medidor 3
<b>Geekflare Tools</b>	No es posible acoplarlo ya que es una herramienta independiente.	No es compatible.	No satisface los requisitos de obtención de información.
<b>Site24x7</b>	No es posible acolarlo ya que es un servicio de pago por uso.	No es compatible.	No satisface los requisitos de obtención de información.
<b>Wheregoes</b>	Es posible que se acople al sistema realizando varios cambios y ajustes.	No es compatible.	No satisface los requisitos de obtención de información.
<b>Redirect-checker</b>	No se puede acoplar ya que no posibilita un consumo de api.	No es compatible.	No satisface los requisitos de obtención de información.

## I.3 Entorno de desarrollo de la propuesta de solución

El proceso que se va a desarrollar es para el Directorio de sitios web cubanos perteneciente al proyecto Red CUBA que forma parte de la dirección de proyectos especiales de UCI. En dicha dirección se tienen predefinidos la metodología de desarrollo de software,

herramientas y lenguajes en el desarrollo con vistas a garantizar la compatibilidad de sus soluciones.

### **I.3.1 Metodología de desarrollo de software**

En la actualidad existen diversas metodologías de desarrollo de software que inciden en distintas dimensiones del proceso de desarrollo, el Instituto Nacional de Tecnologías de la Comunicación de la República de España define a esta metodología como “el conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo” (Herrera, 2009).

Existen actualmente varias metodologías divididas en dos grupos: las tradicionales y las ágiles.

Las metodologías ágiles, caracterizadas por ser más orientadas al desarrollo del software, con bajos niveles de formalización en la documentación requerida y por ser, a diferencia de las tradicionales, más adaptables a los cambios, requerir de pequeños grupos de trabajo y por ser apropiadas para entornos volátiles (Amaro Calderón y Valverde Rebaza, 2007).

Según las especificaciones brindadas por el cliente y teniendo en cuenta la premura de tiempo en que se debería construir el software, se tomó la decisión de utilizar un modelo de desarrollo de tipo ágil. Estos métodos permiten la implementación del software con bajos niveles de formalización en la documentación, en el menor tiempo posible y teniendo al cambio como una constante.

Se selecciona la metodología Programación Extrema (XP) puesto a que independientemente de ser una de las más exitosas en la actualidad, posee una documentación abundante. Es especialmente adecuada para proyectos pequeños con posibilidades de cambio durante el desarrollo. Esta metodología es eficiente, de bajo riesgo, flexible y reduce el tiempo de implementación, por lo que es necesario enfocarse en la programación para obtener un producto funcional en un breve intervalo de tiempo.

#### **Metodología de Programación Extrema (XP)**

La metodología XP o Programación Extrema es una metodología ágil y flexible utilizada para la gestión de proyectos. Extreme Programming se centra en potenciar las relaciones interpersonales del equipo de desarrollo como clave del éxito mediante el trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo. Esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes (Calvo, 2018).

Características:

- Se considera al equipo de proyecto como el principal factor de éxito del proyecto.
- Software que funciona por encima de una buena documentación.
- Interacción constante entre el cliente y el equipo de desarrollo.
- Planificación flexible y abierta.
- Rápida respuesta a cambios.

Roles:

- Cliente: responsable de definir y conducir el proyecto, así como sus objetivos.
- Programadores: Estiman tiempos de desarrollo de cada actividad y programan el proyecto.
- Tester: Encargado de Pruebas.
- Tracker: Encargado de Seguimiento.
- Coach: Entrenador. Su papel es guiar y orientar al equipo.
- Big Boss: Gestor del proyecto, gerente del proyecto, debe tener una idea general del proyecto y estar familiarizado con su estado.

Para el desarrollo del proceso de detección de redirecciones la metodología idónea es XP por ser una metodología ágil muy recomendable para proyectos a corto plazo. Tiene además como ventaja, la programación organizada y planificada, por ser muy eficiente durante el proceso de pruebas y planificación, y por facilitar los cambios que puedan ocurrir durante el desarrollo.

### I.3.2 Lenguajes

#### Java 11.0

Java es un lenguaje de programación y plataforma de computación lanzado por primera vez por Sun Microsystems en 1995. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado, y más se crean todos los días. Java es rápido, seguro y confiable. Desde computadoras portátiles a centros de datos, consolas de videojuegos a supercomputadoras científicas, teléfonos celulares a Internet (*Oracle*, 2018).

Este lenguaje será muy útil ya que es un lenguaje sobre el cual corren la mayoría de los proyectos de la Dirección de proyectos especiales, además de ser muy simple y legible que otros lenguajes dará una mayor facilidad a la hora de corregir los errores que surjan durante el desarrollo de la aplicación.

## **UML**

UML (Unified Modeling Language) o Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos y describe la semántica esencial de estos diagramas y los símbolos en ellos utilizados (Lapuente, 2013). Este lenguaje resultó de gran utilidad durante el desarrollo de la aplicación, pues con él se realizó el diseño de los diagramas, específicamente en el diseño del modelo conceptual y el diagrama de clases y componentes.

### **I.3.3 Herramientas y tecnologías**

#### **Visual Paradigm 8.0**

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de Sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos.

Visual Paradigm for UML Enterprise Edition (VP-UML EE): Es la edición top de la línea de productos, lo que representa lo más moderno y agrega valor en términos de modelado de datos orientado a objetos, hace posible la documentación del proyecto, mapeo relacional de objetos para Java, .NET y PHP, reduciendo costos y aumentando su productividad (Marcotte, 2010).

Posibilita la representación gráfica de los diagramas permitiendo ver el proceso desde diferentes perspectivas como: componentes, despliegue, clases, modelo de datos entre otros. Además, identifica requisitos y comunica información, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, además permite ver las relaciones entre los componentes del diseño.

#### **MySQL 8.0**

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más

populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL es el gestor de base de datos que utiliza el proceso de detección de redirecciones para guardar los datos de los sitios web una vez procesados.

### **Spring Boot 2.7.0**

Spring Boot es una herramienta que nace con la finalidad de simplificar aún más el desarrollo de aplicaciones basadas en el ya popular framework Spring Core. Spring Boot busca que el desarrollador solo se centre en el desarrollo de la solución, olvidándose por completo de la compleja configuración que actualmente tiene Spring Core para poder funcionar.

Spring Boot es adecuado tanto para servicios pequeños como para aplicaciones monolíticas. Estos servicios o aplicaciones pueden ser API web puras o aplicaciones web, línea de comandos programas u orquestación de aplicaciones (Simons, 2020).

### **Git**

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

Esta herramienta es usada en el proyecto, específicamente el Gitlab de la universidad para llevar un registro del control de versiones de los códigos fuentes de las distintas aplicaciones y procesos.

### **Netbeans 12.1**

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE1 es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos (Actualmente Sun Microsystems es administrado por Oracle Corporation) (Domínguez Dorado, 2016).

Este IDE tiene como ventaja que en puede ejecutar en distintos sistemas operativos, es de código abierto y gratuito y permite desarrollar aplicaciones en varios lenguajes.

### **RabbitMQ**

Es un sistema de código abierto que funciona como un middleware de mensajería eficiente que sirve como intermediario entre productores y consumidores, en nuestro caso el monitor de sitios web y el sistema de detección de espejos a desarrollar. Implementa un protocolo de capa de aplicación AMQP (Protocolo de mensaje avanzado de queueing) el cual está enfocado en la comunicación de mensajes asíncronos con garantía de entrega. En otras palabras, RabbitMq define las colas que van a almacenar los sitios web indexados al monitor hasta que la aplicación del detector los analice.

### **Conclusiones del capítulo**

Luego de desarrollar el capítulo se ha podido arribar a las siguientes conclusiones parciales: El análisis de los antecedentes y aspectos teóricos relacionados con el proceso de detección de redirecciones posibilitó un mejor entendimiento del tema investigado.

El estudio de sistemas homólogos, con el objetivo de identificar tendencias actuales relacionadas con las funcionalidades y mecanismos utilizados, permitió identificar las características principales de las herramientas de detección de redirecciones.

El estudio de las tecnologías, herramientas, lenguajes y metodologías definidas por el proyecto permitió identificar sus potencialidades para el desarrollo de esta investigación.

## **CAPÍTULO II: PLANIFICACION Y DISEÑO DE LA SOLUCIÓN**

El desarrollo del presente capítulo está dedicado a la identificación de las principales funcionalidades de la solución propuesta. Durante los epígrafes que componen el mismo, se expondrán los artefactos generados por la puesta en práctica de la metodología de desarrollo de software XP en sus fases de Exploración, Planificación y Diseño. Aborda la solución al problema de investigación descrito en el capítulo anterior.

### **II.1 Propuesta de solución**

Dada la situación problemática planteada en el presente trabajo de diploma se tiene como propuesta de solución desarrollar una aplicación que le permitirá al directorio de sitios web cubano detectar las redirecciones de los sitios web indexados en él, es decir realizar un proceso de detección de redirecciones.

El proceso comenzará obteniendo la URL de cada sitio que se quiera indexar en el directorio de sitios web y generar una tabla de rutas con el estado de cada petición para tener el inicio y el fin de la URL de un sitio. Así cuando se inserte un sitio se pueda saber si es una redirección o no la URL que se está adicionando. El primer paso sería consultar el código de estado de cada una de las URL donde se va a centrar en 3xx que tratan sobre la redirección. Si el código de estado es 301, 302 o 307 es porque existe una redirección. El paso siguiente es registrar en la tabla la fecha en que se realiza la consulta, la URL inicial, el estado que devolvió y la URL final, y si es una redirección repetir el proceso hasta que el código de estado de la URL sea 200, siendo esta la URL final.

#### **II.1.1 Modelo conceptual**

Un modelo conceptual (ver Figura 2) es una representación de conceptos en un dominio del problema. La designación de modelo conceptual ofrece la ventaja de subrayar fuertemente una concentración en los conceptos del dominio, no en las entidades del software (Larman, 2003).

El modelo conceptual relaciona las entidades que intervienen en el proceso de la detección de redirecciones, estas son:

- Los sitios web que serán analizados por el proceso de detección de redirecciones.
- El proceso de detección de redirecciones que es el más importante pues se encargará de realizar el análisis de los sitios web.

- La Entidad es la base de datos encargada de almacenar la información de los sitios web analizados para saber si poseen redirección o no.
- El directorio de sitios web que es quien recibe los sitios web previamente procesados.

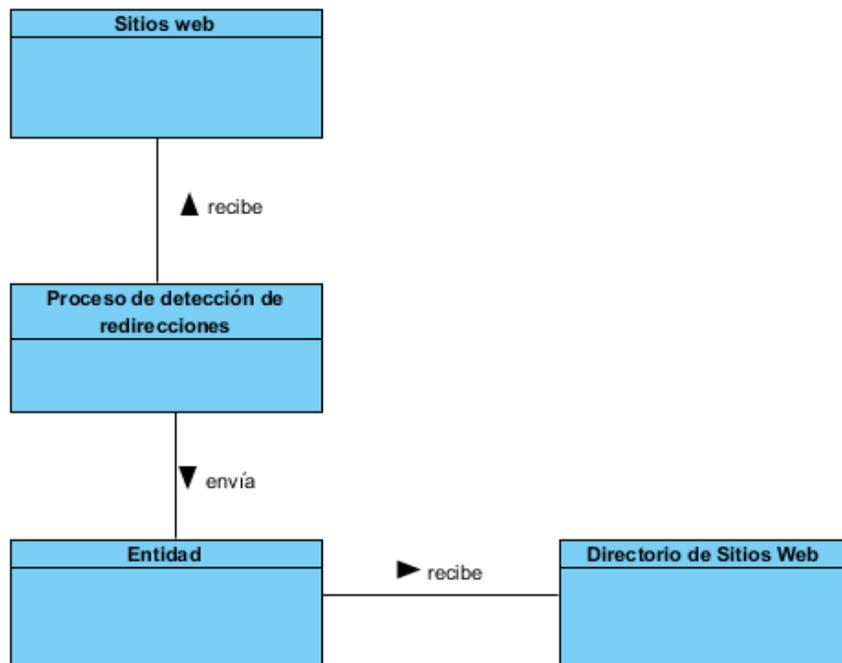


Figura 2: Modelo conceptual. Fuente: (Elaboración propia)

### II.1.2 Requisitos funcionales

Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (Somerville, 2005).

RF 1: Detectar redirección a partir de una URL.

RF 2: Detectar código de estado de la URL.

RF 3: Insertar fecha en la que se realiza la consulta a la URL.

RF 4: Verificar si el sitio posee una redirección.

RF 5: Insertar valores (fecha, URL inicial, código de estado y URL final) en la base de datos.

RF 6: Desarrollar un endpoint para conocer si el sitio en cuestión es una redirección a través del código de estado que presenta.

### II.1.3 Requisitos no funcionales

Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente se aplican en poca medida a características o servicios individuales del sistema (Somerville, 2005).

A continuación, se muestran los requisitos no funcionales que presenta la Aplicación para de detección de redirecciones en el Directorio de Sitios Web:

- Usabilidad:
  1. RNF 1: Permite la detección de redirecciones a partir de ser detectado un sitio web, realizando consultas a su código de estado.
- Eficiencia:
  - 1 RNF 2: El tiempo de respuesta de una petición al servidor debe ser menor a cinco segundos.
- Portabilidad:
  - 1 RNF 3: Se puede utilizar la aplicación en sistemas operativos Windows y GNU/Linux. Se recomienda utilizar los de GNU/Linux.
- Disponibilidad:
  - 1 RNF 4: La disponibilidad de la aplicación debe ser continua con un nivel de servicio para los usuarios de siete días por 24 horas.

### II.1.3 Historias de usuarios

Las Historias de Usuario representan una breve descripción del comportamiento del sistema, se realizan por cada característica principal del sistema y son utilizadas para cumplir estimaciones de tiempo y el plan de lanzamientos, así mismo reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación. Cada historia de usuario debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarlas en unas semanas (Meléndez Valladares et al., 2016).

Las HU son representadas mediante tablas divididas por las siguientes secciones:

- **Número:** Permite identificar a una historia de usuario.
- **Usuario:** Persona que utilizará la funcionalidad del sistema descrita en la historia de usuario.
- **Nombre de Historia de Usuario:** Describe de manera general a una historia de usuario.
- **Prioridad en negocio:** grado e importancia que el cliente asigna a una historia de usuario.
- **Riesgo en Desarrollo:** Valor de complejidad que una historia de usuario representa al equipo de desarrollo.
- **Puntos Estimados:** Número de semanas que necesitara para el desarrollo de una historia de usuario.
- **Iteración Asignada:** Número de iteración, en que el cliente desea que se implemente una historia de usuario.
- **Programador Responsable:** Persona encargada de programar cada historia de usuario.
- **Descripción:** Información detallada de una historia de usuario.
- **Observaciones:** Campo opcional utilizado para aclarar, si es necesario, el requerimiento descrito de una historia de usuario.

A continuación, se muestran las historias de usuario que describen el proceso de detección de redirecciones en el Directorio de Sitios Web (ver tablas de la 3 a la 8).

Tabla 3: HU1

Historia de Usuario	
<b>Número:</b> 1	<b>Usuario:</b> Desarrollador
<b>Nombre de Historia:</b> Detectar redirección a partir de una URL.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos Estimados:</b> 1	<b>Iteración Asignada:</b> 1
<b>Programador Responsable:</b> Iris Leybis Atencio Ramírez	
<b>Descripción:</b> El proceso detecta a partir de una URL dada si tiene redirección a otra URL o no.	

Tabla 4: HU2

Historia de Usuario	
<b>Número:</b> 2	<b>Usuario:</b> Desarrollador
<b>Nombre de Historia:</b> Detectar código de estado de la URL.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos Estimados:</b> 1	<b>Iteración Asignada:</b> 1
<b>Programador Responsable:</b> Iris Leybis Atencio Ramírez	
<b>Descripción:</b> Se obtiene el código de estado de la URL.	

Tabla 5: HU3

Historia de Usuario	
<b>Número:</b> 3	<b>Usuario:</b> Desarrollador
<b>Nombre de Historia:</b> Insertar fecha en la que se realiza la consulta a la URL	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 1
<b>Programador Responsable:</b> Iris Leybis Atencio Ramírez	
<b>Descripción:</b> Se inserta la fecha de cuando se consulta el código de estado de la URL.	

Tabla 6: HU4

Historia de Usuario	
<b>Número:</b> 4	<b>Usuario:</b> Desarrollador
<b>Nombre de Historia:</b> Verificar si el sitio posee una redirección.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos Estimados:</b> 3	<b>Iteración Asignada:</b> 1
<b>Programador Responsable:</b> Iris Leybis Atencio Ramírez	
<b>Descripción:</b> El proceso verifica si la URL dada, a partir de su código de estado posee una redirección o no.	

Tabla 7: HU5

Historia de Usuario	
<b>Número:</b> 5	<b>Usuario:</b> Desarrollador
<b>Nombre de Historia:</b> Insertar valores (fecha, URL y código de estado) en la base de datos.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 2
<b>Programador Responsable:</b> Iris Leybis Atencio Ramírez	
<b>Descripción:</b> Se Insertan los valores Fecha, URL y Código de estado en la tabla de la base de datos.	

Tabla 8: HU6

Historia de Usuario	
<b>Número:</b> 6	<b>Usuario:</b> Desarrollador
<b>Nombre de Historia:</b> Desarrollar un endpoint para conocer si el sitio en cuestión es una redirección a través del código de estado que presenta.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos Estimados:</b> 3	<b>Iteración Asignada:</b> 3
<b>Programador Responsable:</b> Iris Leybis Atencio Ramírez	
<b>Descripción:</b> Con la implementación de este endpoint se evalúa si el sitio web analizado es una redirección o no.	

### II.1.4 Plan de iteraciones

Todo proyecto que siga la metodología XP se ha de dividir en iteraciones de aproximadamente tres semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las HU que serán implementadas.

A continuación, se muestra el Plan de iteraciones relacionado con el desarrollo del proceso de detección de redirecciones (ver tabla 9).

Tabla 9: Plan de iteraciones.

Iteración	Duración (Semanas)	Historias de Usuarios
1	3	1-4
2	3	5
3	3	6

**Iteración 1:** En esta iteración se implementarán las HU de la 1 a la 4 que están relacionadas con la primera etapa del proceso de detección de redirecciones. Se realizará la exploración pertinente a los conceptos asociados a la investigación a desarrollar, se definirán las metodologías de desarrollo de software, así como los lenguajes y herramientas a utilizar para dar solución a la problemática.

**Iteración 2:** Esta iteración se corresponde con la HU número 5 con una prioridad alta en el negocio donde se insertarán en la base de datos los valores involucrados en el proceso de detección de redirecciones, que son la fecha de consulta, la URL y el código de estado de la URL.

**Iteración 3:** En esta iteración se implementará la HU número 6, se realizarán las pruebas al proceso y se someterá a criterios de expertos para así asegurar su correcto funcionamiento. Al concluir estas pruebas con resultados satisfactorios se obtendrá una versión casi culminada.

## **II.2 Diseño**

En esta fase se realiza el diseño del sistema. Con las facilidades que brinda la metodología XP la estrategia de diseño es simple, con el menor número de clases y métodos posibles, lo que propicia su fácil implementación con menos tiempo y esfuerzo.

### **II.2.1 Arquitectura de software**

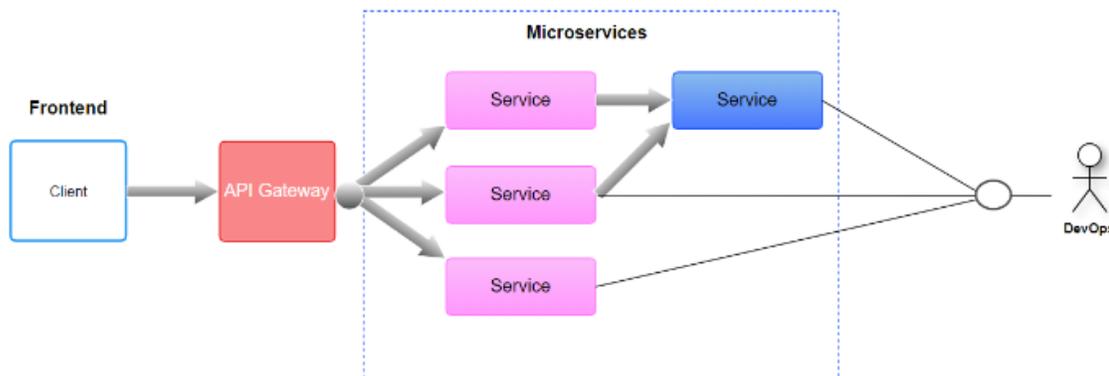
La arquitectura de software de un sistema de programa o computación es la estructura del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos (Pressman, 2010).

La arquitectura representa la clave para comprender, organizar y comunicar un sistema, además, facilita la evolución de la solución. Es diseñada para satisfacer los requerimientos funcionales y no funcionales establecidos por los usuarios, clientes y proveedores del sistema. Un software que no posee un correcto diseño arquitectónico puede funcionar de forma muy deficiente o simplemente no funcionar generando consecuencias para la organización que sirve. Para las empresas que dependen de los sistemas de información las arquitecturas de software son fundamentales para el logro de sus objetivos organizacionales, lo que incluye el poder evolucionar rápidamente según las condiciones altamente cambiantes de los mercados actuales (Villegas Ballesta, 2015).

### **Arquitectura Microservicios**

Los micro servicios son un tipo de arquitectura que sirve para diseñar aplicaciones. Lo que distingue a la arquitectura de micro servicios de los enfoques tradicionales y monolíticos es la forma en que desglosa una aplicación en sus funciones principales. Cada función se denomina servicio y se puede diseñar e implementar de forma independiente. Esto permite que funcionen separados sin afectar a los demás. Las aplicaciones complejas se componen de procesos independientes pequeños que se comunican entre sí mediante API que no utilizan el mismo idioma (Vargas González, 2019).

Esta arquitectura es una forma de desarrollo software que se basa en la implementación independiente de cada uno de los servicios de la aplicación. En la Figura 3 se encuentra representada la representación del diseño de esta arquitectura.



**Figura 3** Arquitectura de microservicios. Fuente (Diego Navarro y Cabrera Lozada, 2020)

Como se puede observar en la figura, hay dos tipos de conexiones en esta arquitectura, la primera conecta la interfaz (frontend) con los microservicios a través de una puerta de enlace (Gateway) de API (Application Programming Interface), y la segunda es cualquier conexión entre los servicios que poseen la lógica de negocio, por ejemplo, la que relaciona una base de datos con una aplicación Java (Diego Navarro y Cabrera Lozada, 2020).

Para el desarrollo de la solución se utilizó la arquitectura de microservicios que es la que más se ajusta al desarrollo de la aplicación por ser esta una funcionalidad muy específica para un sistema que es el Directorio de Sitios Web cubanos que funciona de forma desagregada y que tiene múltiples componentes que emplean una arquitectura muy similar.

### II.2.2 Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. En un diagrama de clases se pueden distinguir principalmente dos elementos: clases y sus relaciones (Flores Cueto y Bertolotti Zuñiga, 2015).

El diagrama de clases de solución propuesta que aparece a continuación (ver Figura 4) aporta una mejor comprensión del funcionamiento del proceso, detallando cada una de las clases que intervienen en él y sus relaciones.

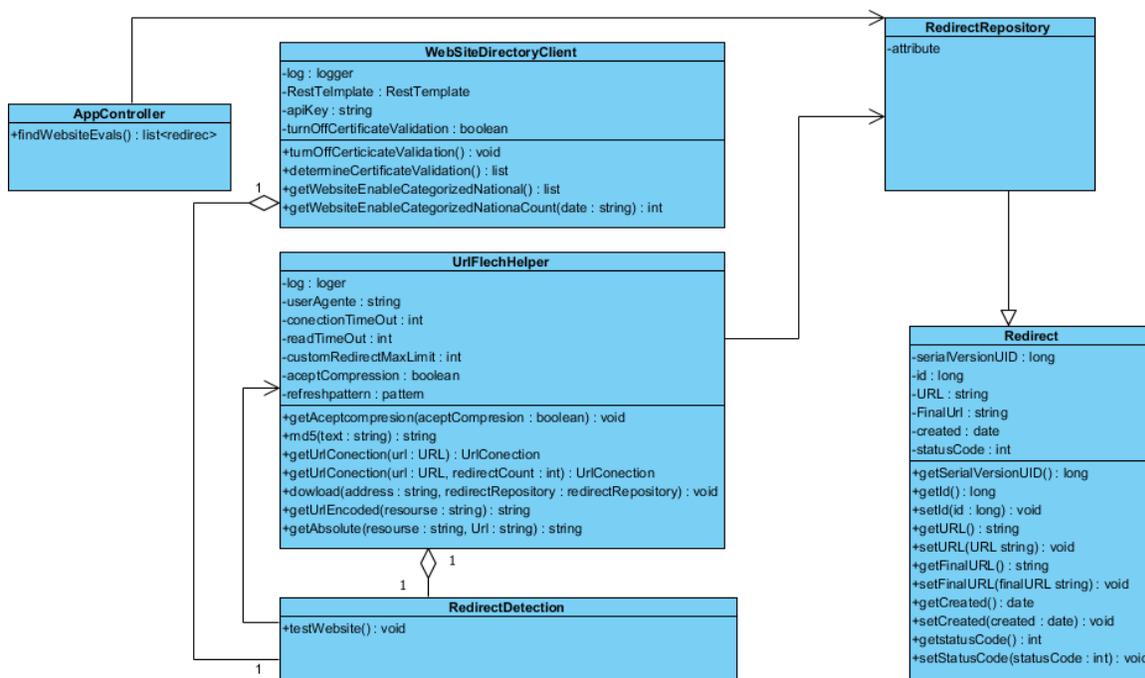


Figura 4: Diagrama de clases. Fuente: (Elaboración propia)

### II.2.3 Patrones de diseño

Un patrón es una descripción de un problema y la solución, a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos (Larman, 2003).

#### II.2.3.1 Patrones de diseño GRASP

Para el diseño de la propuesta de solución se tuvieron en cuenta los patrones Generales de Software para Asignación de Responsabilidades GRASP (General Responsibility Assignment Software Patterns), los Patrones Generales de Software para Asignar Responsabilidades describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. Es importante destacar que estos patrones están estrechamente relacionados entre sí.

## Creador

Este patrón plantea que se debe asignar a una clase B la responsabilidad de crear una instancia de una clase A (Larman, 2003). La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. Este patrón es el encargado de guiar la asignación de responsabilidades relacionadas con la **creación de objetos**. Este patrón se evidencia el uso del patrón creador, al crear las variables `matcher` y `next` (ver Figura 5).

```
String refresh = http.getHeaderField( name: "Refresh");
if (null != refresh && !"".equals(refresh.trim())) {
    Matcher matcher = refreshPattern.matcher(refresh);
    if (matcher.matches()) {
        http.disconnect();
        // Obtener la dirección.
        URL next = new URL(matcher.group(1).trim());
    }
}
```

Figura 5: Ejemplo de patrón creador. Fuente: (Elaboración propia)

## Alta cohesión

Este patrón plantea que se debe asignar una responsabilidad de modo que la cohesión siga siendo alta (una clase tiene responsabilidades moderadas). Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que no realicen un trabajo enorme. Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo. Esto se evidencia en la creación de la clase `RedirectDetection`.

## Experto

Es el encargado de asignar una responsabilidad al experto en información: la clase que tiene la información necesaria para realizar la responsabilidad. Indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. Este patrón se encuentra representado en la clase `UrlFletchHelper`.

## Bajo acoplamiento

Acoplamiento indica que tan fuertemente está conectada una clase con otra, tiene conocimiento de, o influye sobre otra clase, una clase con bajo acoplamiento no depende de otras clases. El bajo acoplamiento soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio (Larman, 2003).

Este patrón se manifiesta en la clase `WebSiteDirectoryClient` que es la clase que representa los servicios del directorio. Esta clase no depende de otra clase en la aplicación y esto permite que en caso de modificarse algún elemento sus cambios no afecten a otro componente.

### **Controlador**

El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es el controlador quien recibe los datos del usuario y quien los envía a las distintas clases según el método llamado (García Carmona, s.f.). La clase que implementa el patrón controlador es la `Appcontroller` que es la clase controladora principal de datos.

### **II.2.3.1 Patrones de diseño GOF**

Los patrones de diseño GOF son descripciones de objetos que se comunican y clases que son personalizadas para resolver un problema de diseño general en un contexto particular. Estos patrones se clasifican según su propósito en creacionales, estructurales y de comportamiento, mientras que respecto a su ámbito se clasifican en clases y objetos. Los empleados en la solución son los creacionales.

**Builder (constructor virtual):** Separar la construcción de un objeto complejo de su representación para que un mismo proceso de creación pueda crear diferentes representaciones. Se evidencia en la clase `UrlFetchHelper`.

**Singleton (instancia única):** garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Restringe la instanciación de una clase o valor de un tipo a un solo objeto. Se evidencia en la clase `WebSiteDirectoryClient`.

**Factory Method (método de fabricación):** centraliza en una clase constructora la creación de objetos de un subtipo de un tipo determinado. Se evidencia en la clase `UrlFetchHelper`.

### **II.2.3 Tarjetas Clase-Responsabilidad-Colaborador**

Una tarjeta CRC es en realidad una colección de tarjetas índices estándar que representan clases. Estas se dividen en tres secciones, generalmente el nombre de la clase se coloca en el borde superior en forma de título, en la parte derecha los colaboradores (clases que

se implican en cada funcionalidad) y las responsabilidades (funcionalidades) de la clase en el extremo izquierdo (Pressman, 2010).

A continuación, se muestran las tarjetas CRC que intervienen en el proceso de detección de redirecciones (ver tablas de la 10 a la 15).

Tabla 10: Tarjeta CRC1

Tarjeta CRC 1	
<b>Clase:</b> ApplicationController	
<b>Responsabilidades:</b> Representa el controlador principal de datos.	<b>Colaboraciones:</b> RedirecRepository Redirect

Tabla 11: Tarjeta CRC2

Tarjeta CRC 2	
<b>Clase:</b> WebSiteDirectoryClient	
<b>Responsabilidades:</b> Representa los servicios del Directorio de Sitios Web. Cliente que se comunica con el directorio.	<b>Colaboraciones:</b> N/A

Tabla 12: Tarjeta CRC3

Tarjeta CRC 3	
<b>Clase:</b> Redirec	
<b>Responsabilidades:</b> Representa la Base de Datos donde se almacenan los datos de los sitios web analizados.	<b>Colaboraciones:</b> N/A

Tabla 13: Tarjeta CRC4

Tarjeta CRC4	
<b>Clase:</b> UrlFletchHelper	
<b>Responsabilidades:</b> Identificar los sitios que poseen una redirección y hace la conexión a un sitio web.	<b>Colaboraciones:</b> RedirecRepository Redirect

Tabla 14: Tarjeta CRC5

Tarjeta CRC 5	
<b>Clase:</b> RedirectRepository	
<b>Responsabilidades:</b> Se encarga de manejar el acceso a la base de datos, clase manejadora donde están las consultas.	<b>Colaboraciones:</b> Redirec

Table 15: Tarjeta CRC6

Tarjeta CRC 6	
<b>Clase:</b> RedirecDetection	
<b>Responsabilidades:</b> Se define cada que tiempo será ejecutado el proceso. Controla el proceso de detección de redirecciones y se conecta con los servicios del directorio.	<b>Colaboraciones:</b> RedirecRepository WebSiteDirectoryClient UrlFlechHelper

### II.3 Modelo de datos

El modelo de datos permite describir las estructuras de la base de datos, el tipo de datos que hay en la base de datos y la forma en que se relacionan.

Para la implementación del proceso de detección de redirecciones es necesario el uso de una nueva base de datos, utilizada para almacenar los datos de los sitios web como la fecha en la que se consulta el sitio, la URL y el código de estado que presenta esa URL (ver Figura 5).

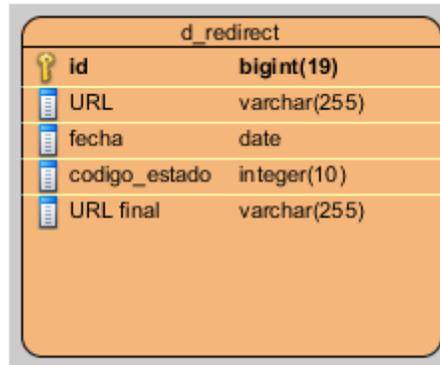


Figura 5 Modelo de datos. Fuente: (Elaboración propia)

### Conclusiones del capítulo

Finalizado este capítulo se pudo concluir que se realizó un análisis y diseño de la propuesta de solución: Para ello se elaboró un modelo conceptual y se identificaron 6 requisitos funcionales y 6 historias de usuario, en las cuales se realizaron las descripciones de dichos requisitos. Mediante la identificación de la arquitectura se pudo lograr una mayor organización de los elementos que darán forma a la aplicación. Se diseñaron el modelo de datos respondiendo a las exigencias y a las clases que posee la solución propuesta, las tarjetas CRC y un diagrama de clases para una mejor comprensión.

## CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Culminada la etapa de diseño, que sienta las bases estructurales del sistema, se inicia la fase de implementación y pruebas para lograr un producto con calidad. Darles seguimiento a los cambios y desarrollar iterativamente. En este capítulo se plasman los casos de pruebas a los que fue sometido el proceso de detección de redirecciones y las pruebas realizadas.

### III.1 Diagrama de componentes

Un diagrama de componentes (UML) representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes (Giraldo G et al., 2011).

Los diagramas de componentes representan las relaciones entre los componentes individuales del sistema mediante una vista de diseño estática. Pueden ilustrar aspectos de modelado lógico y físico.

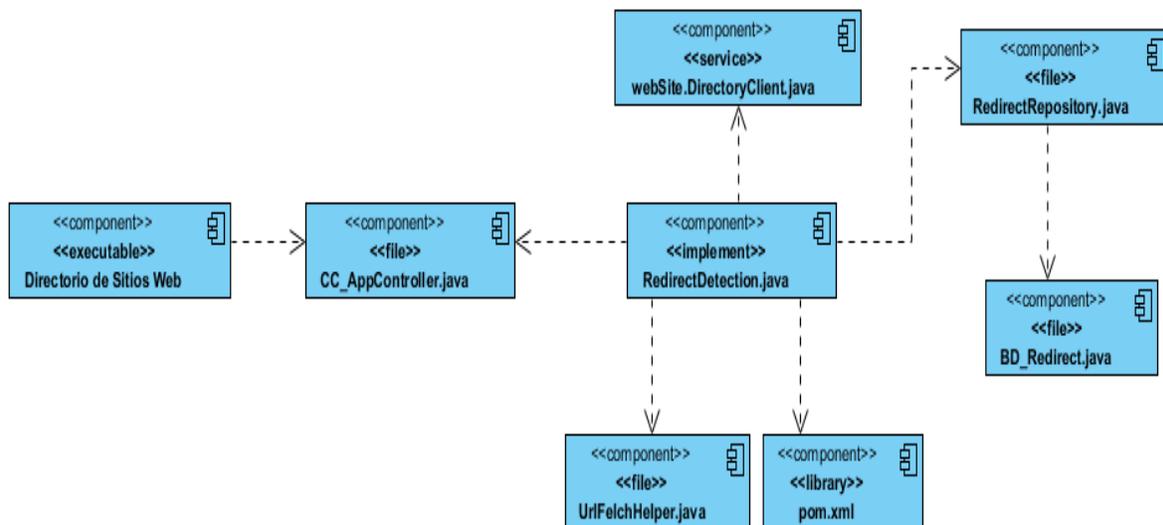


Figura 6 Diagrama de componentes. Fuente: (Elaboración propia)

En el diagrama representado en la Figura 6 se evidencian los componentes por separado que conforman la estructura de la aplicación una vez integrada al Directorio de Sitios Web cubanos. El componente Directorio de Sitios Web representa el cliente para el que se ha desarrollado el microservicio, que es conectado a la solución propuesta mediante un endpoint desde el cual el API puede acceder a los recursos que necesita, representado en

el componente `CC_AppController.java`. `RedirectDetection.java` relacionada con `webSitedirectoryCliente.java` que representa los servicios del directorio, con `UrlFelchHelper.java` que es el que hace la conexión con un sitio web, con las librerías `pom.xml` de las que se usan sus funcionalidades comunes y con `redirectRepository.java` que hace la conexión con la base de datos `BD_Redirect.java` donde van a ser almacenados los resultados del proceso de detección de redirecciones, todos estos componentes representan los servicios.

### III.2 Estándares de codificación

Los estándares de codificación comprenden todos los aspectos de generación de código en un proyecto y permiten entender de manera rápida, fácil y sencilla el código empleado en el desarrollo de un software.

El objetivo de los estándares de codificación de software es inculcar prácticas de programación probadas que conduzcan a un código seguro, confiable, comprobable y mantenible. Por lo general, esto significa evitar las prácticas de codificación inseguras conocidas o el código que puede causar un comportamiento impredecible (Hamilton, 2020). La confección de estos estándares debe ser definida al comienzo de la implementación para garantizar que todos los programadores trabajen de manera coordinada. A continuación, describen algunos de los estándares seguidos para el desarrollo de la solución.

- Utilizar la siguiente estructura de organización del código:
  - Comentario de comienzo.
  - Sentencias de importación de paquetes.
  - Declaración de las clases.
- Codificación UTF-8 en todos los módulos.
- Evitar usar espacios en blanco innecesarios.
- Cada línea de código no debe exceder los 80 caracteres en la medida de lo posible.
- Las variables declaradas como contantes deben ir totalmente en mayúsculas con las palabras separadas con una barra baja `_`.
- Las variables de una sola letra deben ser evitadas salvo los casos comúnmente reconocidos como iteradores (`i`, `j`, `k`).
- Usar líneas en blanco para separar secciones, entre definición de clases, entre métodos, entre variables y la primera sentencia.

- Usar un espacio entre una palabra clave (if, for, while) y el paréntesis a continuación. Todos los operadores excepto el punto, los de incremento ++ y decremento -- deben separarse de sus operandos con un espacio.
- Escribir comentarios para encontrar código más fácilmente.
  - Los comentarios deben ser oraciones completas.
  - Si un comentario es corto el punto final puede omitirse.
  - Los comentarios de una línea para aclaraciones del código aparecerán seguidos de los caracteres “//” y deben ubicarse en la misma línea que se desee comentar.

### III.3 Pruebas de software

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un sistema. Básicamente es una fase en el desarrollo de software que consiste en probar la aplicación construida. Se integran dentro de las diferentes fases del ciclo de la Ingeniería de Software ejecutando un programa que mediante técnicas experimentales se trata de descubrir los errores que tiene el sistema. Para determinar el nivel de calidad se deben efectuar las medidas o pruebas que permiten comprobar el grado de cumplimiento respecto a las especificaciones iniciales del producto, siendo el resultado observado y registrado.

El proceso de pruebas es un elemento fundamental de la metodología XP. Este reduce el número de errores no detectados, así como el tiempo entre su introducción en el sistema y su detección. Dicha metodología divide las pruebas en varios grupos entre los que se encuentran:

**Pruebas Unitarias:** Son las pruebas implementadas por los desarrolladores, encargadas de verificar el código.

**Pruebas de Integración:** Son la herramienta que conjunta cada uno de los componentes de un sistema para comprobar su funcionamiento entre sí.

**Pruebas de Aceptación:** Validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación.

### III.3.1 Pruebas unitarias

#### Pruebas de caja blanca

Las pruebas de caja blanca, denominadas a veces pruebas de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba (Pressman, 2010). Con la aplicación de este método se garantiza que se verifique por lo menos una vez todos los caminos independientes de cada módulo, que se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas, que se ejecuten todos los bucles en sus límites y con sus límites operacionales y se entrenen las estructuras internas de datos para asegurar su validez.

La técnica del camino básico fue aplicada un fragmento del código del método getUrlConection, que es el encargado de verificar a través de consultas al código de estado de la URL si el sitio es una redirección (ver Tabla 16).

Tabla 16: Cálculo de la complejidad ciclomática del fragmento del método getUrlConection.

<b>Fragmento de código:</b>		
<pre>String location = http.getHeaderField( name: "Location"); if (null != location &amp;&amp; !"".equals(location.trim())) {     http.disconnect();     // Lidiar con URLs relativas.     URL next = new URL(url, location);     return getUrlConection(next, ++redirectCount); } break;</pre>		
<b>Grafo resultante:</b>		
<pre> graph TD     1((1)) --&gt; 2((2))     2 --&gt; 3((3))     3 --&gt; 4((4))     3 --&gt; 5((5))     4 --&gt; 5     5 --&gt; 5     </pre>		
<b>Complejidad ciclomática:</b>	$V(G) = A - N + 2$	$V(G) = P + 1$
$V(G) = \# \text{ de regiones}$	$V(G) = 6 - 5 + 2$	$V(G) = 2 + 1$
$V(G) = 3$	$V(G) = 3$	$V(G) = 3$

Luego de la determinación de los nodos y flujos de control del código se obtuvo el grafo de flujo y se calculó la complejidad ciclomática del algoritmo.

Como resultado se obtuvo que la complejidad ciclomática es igual a 3, lo que significa que existen tres posibles caminos linealmente independientes y hay que diseñar un mínimo de tres casos de prueba para el fragmento del método getUrlConexion. La tabla 17 muestra los caminos existentes.

**Tabla 17: Caminos del grafo de flujo.**

No.	Camino
1	1,2,3,4,5
2	1,2,3,5
3	1,2,5

Los casos de prueba para las pruebas de caja blanca por la técnica de camino básico se ejecutan por cada camino independiente que se determine en un algoritmo específico. A continuación (ver tabla de la 18 a la 20), se muestran los casos de prueba para los tres caminos detectados en el fragmento de método: “Detectar si el protocolo redirigido es el mismo”.

**Tabla 18: Caso de prueba Protocolo redirigido. EC1.1.**

<b>Proceso:</b>
Detectar si el protocolo redirigido es el mismo que el de la URL pasada por parámetros.
<b>Caso de prueba:</b>
Protocolo redirigido. Escenario 1.1
<b>Camino independiente:</b>
1,2,3,4,5
<b>Entradas:</b>
Se crea la variable location inicializada con un valor igual a una URL de tipo http, location debe ser distinto de null, a la variable location se le eliminan los espacios en blanco de su inicio y de su fin, esta toma un valor distinto de vacío y entra al bloque de ejecución.
<b>Resultados:</b>
Se termina el proceso.
<b>Condiciones:</b>
Que exista al menos una URL.

Tabla 19: Caso de prueba Protocolo redirigido. EC1.2.

<b>Proceso:</b>
Detectar si el protocolo redirigido no es el mismo que el de la URL pasada por parámetros.
<b>Caso de prueba:</b>
Protocolo redirigido. Escenario 1.2
<b>Camino independiente:</b>
1,2,3,5
<b>Entradas:</b>
Se crea la variable location inicializada con un valor igual a una URL de tipo http, location debe ser distinto de null, a la variable location se le eliminan los espacios en blanco de su inicio y de su fin, esta toma un valor distinto igual a vacío.
<b>Resultados:</b>
Se termina el proceso.
<b>Condiciones:</b>
Que exista al menos una URL.

Tabla 20: Caso de prueba Protocolo redirigido. EC1.3.

<b>Proceso:</b>
Detectar si el protocolo redirigido no es el mismo que el de la URL pasada por parámetros.
<b>Caso de prueba:</b>
Protocolo redirigido. Escenario 1.3
<b>Camino independiente:</b>
1,2,5
<b>Entradas:</b>
Se crea la variable location inicializada con un valor igual a una URL de tipo http, location es igual a null.
<b>Resultados:</b>
Se termina el proceso.
<b>Condiciones:</b>
Que exista al menos una URL.

### III.3.2 Pruebas de integración

Las pruebas de integración están diseñadas para probar la interacción entre los distintos componentes de un sistema, también pueden certificar el funcionamiento o integración entre dos o más sistemas. Su objetivo es tomar los componentes probados en unidad y construir una estructura acorde a la que propone el diseño (*Pruebas de integración*, 2013).

Para probar la correcta integración de los microservicios desarrollados se realizaron procesos de evaluación a los sitios web detectados por los rastreadores. En cada uno de los casos el sistema recibe correctamente los datos de los sitios web, realiza el proceso de detección de redirecciones y su almacenamiento en las bases de datos de manera satisfactoria (ver tabla 21). Por lo que se puede constatar que los microservicios están listos para integrarse al Directorio de Sitios Web cubanos.

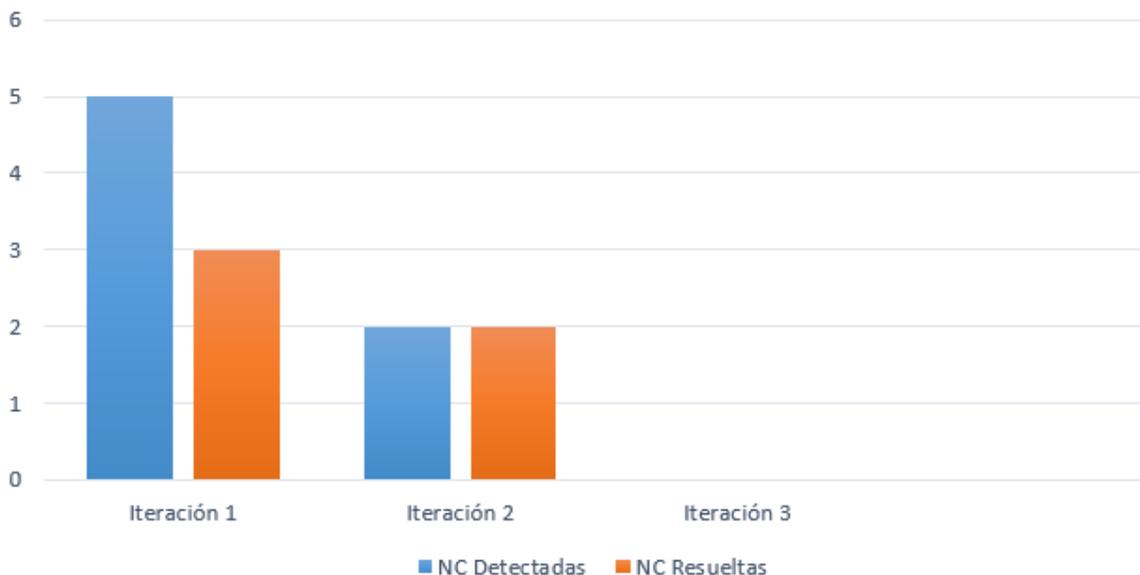
**Tabla 21 Caso de prueba integración de la solución.**

<b>Escenario</b>	<b>Descripción</b>	<b>Manual de Configuración</b>	<b>Valor Esperado</b>	<b>Respuesta de la Aplicación</b>
El sistema detecta un nuevo sitio web.	El sistema realiza la detección de redirecciones y envía la información a la base de datos, para posteriormente mostrarla en la aplicación Directorio de Sitios Web cubanos.	El sitio web tiene que estar disponible antes de realizar la evaluación.	1 El sitio posee una redirección. 2 El sitio no posee redirección.	El sistema verifica que el sitio esté disponible, para posteriormente clasificarlo, almacenarlo en la base de datos y mostrarlo en la aplicación Directorio de Sitios Web cubanos.

### Resultados de las pruebas de integración:

Para llevar a cabo la integración de la Aplicación para la detección de redirecciones con el Directorio de Sitios Web se realizaron dos acciones:

1. Integración de la solución desarrollada con el Directorio de sitios Web.
2. Comprobación de la integración y rectificación de no conformidades.



**Figura 7 Resultados de las pruebas de integración. Fuente (Elaboración propia)**

Para llevar a cabo las pruebas de integración se realizaron tres iteraciones donde se obtuvieron varias no conformidades. En la Iteración 1 se identificaron cinco no conformidades, resolviéndose 3 de ellas. En la Iteración 2 se detectaron dos no conformidades las cuales fueron resueltas y en la Iteración 3 no se detectó ninguna no conformidad.

### III.3.3 Pruebas de aceptación

La prueba de aceptación es generalmente desarrollada y ejecutada por el cliente o un especialista de la aplicación y es conducida a determinar cómo el sistema satisface sus criterios de aceptación, validando los requisitos que han sido levantados para el desarrollo, incluyendo la documentación y procesos de negocio. Está considerada como la fase final del proceso, para crear un producto confiable y apropiado para su uso (Pressman, 2010).

Para llevar a cabo estas pruebas de aceptación existen dos tipos de pruebas la Alfa y Beta. El tipo de prueba Alfa depende completamente de la información proporcionada por los usuarios finales, en relación a su experiencia con el producto de software. Por tal motivo,

los desarrolladores les proporcionan las aplicaciones, la página web o el sistema desarrollado para que ellos interactúen con él y emitan un informe sobre el mismo. Quienes la ejecutan suele ser personal interno o de confianza a la organización.

Para la aplicación de esta prueba se seleccionó un grupo de especialistas de la dirección de Proyectos Especiales y se hizo entrega por parte de los clientes de un aval que certifican su satisfacción con la aplicación realizada (ver Figura 8).

**UCI** Universidad de las Ciencias Informáticas

## Acta de Aceptación

Acta de Aceptación

En el cumplimiento del Convenio de colaboración establecido entre el centro Dirección de Proyectos Especiales y la estudiante Iris Leybis Atencio Ramírez de la Facultad 1, en la Universidad de las Ciencias Informáticas y en función de la ejecución del Proyecto de Tesis: Aplicación para la detección de redirecciones en el Directorio de Sitios Web. Se hace entrega del proyecto que se relaciona a continuación:

- ❖ Aplicación para la detección de redirecciones en el Directorio de Sitios Web.

La parte cliente, luego de haber revisado el producto de trabajo relacionado anteriormente, procede a firmar la aceptación del mismo en total conformidad.

Entrega	Recibe
<p><i>Iris Leybis Atencio Ramírez</i> Nombre y apellidos:</p> <p><i>estudiante</i> Cargo:</p> <p><i>[Firma]</i> Firma:</p>	<p><i>José Gabriel Espinosa R.</i> Nombre y apellidos:</p> <p><i>Jefe de proyecto SEO</i> Cargo:</p> <p><i>[Firma]</i> Firma:</p>

Figura 8 Aval de aceptación de la solución.

### III.4 Diagrama de despliegue

No es más que en un diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista del despliegue o distribución de los artefactos del software en los destinos en que se propone el mismo. Define a los artefactos como representaciones de elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo. Ejemplos de artefactos son los archivos ejecutables, bibliotecas, archivos, esquemas de base de datos y archivos de configuración. Cuando se hace mención de destino de despliegue no es más que un nodo el cual es un dispositivo de hardware o un entorno de ejecución de software (Sarmiento, 2013).

Para lograr una representación de la distribución física de la aplicación para la detección de redirecciones en el Directorio de Sitios Web cubanos, se muestra en la Figura 9 el diagrama de despliegue correspondiente.



Figura 9 Diagrama de despliegue. Fuente: (Elaboración propia)

Este diagrama se considera para lograr un despliegue exitoso de la aplicación y se define un servidor central donde se encuentra la base de datos y la solución implementada.

### Conclusiones del capítulo

En este capítulo se definieron un total de 9 estándares de codificación para la implementación facilitando un proceso funcional capaz de cumplir con los objetivos propuestos. Se validó la propuesta de solución a través de estrategias de pruebas que facilitó corroborar la calidad de la misma, estableciendo así el cumplimiento de los requisitos definidos por el cliente.

## **CONCLUSIONES GENERALES**

Durante el desarrollo de la investigación se planteó la necesidad de implementar una aplicación para la detección de redirecciones en el directorio de sitios web, dándole así cumplimiento al objetivo propuesto de la siguiente forma:

- El estudio de los referentes teóricos permitió identificar las características principales del proceso, propiciando un mejor entendimiento del diseño de la propuesta de solución.
- El análisis del estado del arte, las tendencias actuales del uso de las redirecciones y del posicionamiento web permitió ver la necesidad existente de desarrollar una aplicación para la detección de redirecciones en el directorio de sitios web.
- La implementación del proceso de detección de redirecciones en el Directorio de Sitios Web contribuye a el posicionamiento de los sitios web cubanos.
- La estrategia de pruebas permitió validar la calidad de las especificaciones en el código, y finalmente con las pruebas unitarias y de aceptación se lograron resolver las no conformidades que presentaba la aplicación.

## **RECOMENDACIONES**

En función del constante proceso de mejora y evolución que es inherente a todo sistema de software se recomienda lo siguiente:

- Adicionar a la aplicación desarrollada la funcionalidad de detección de réplicas.

## REFERENCIAS BIBLIOGRÁFICAS

- Amaro Calderón, S. D., & Valverde Rebaza, J. C. (2007). *Metodoloías Ágiles*. <https://silo.tips/download/universidad-nacional-de-trujillo-12>
- API Endpoints - What Are They? Why Do They Matter?* (2022). smartbear.com. <https://smartbear.com/learn/performance-monitoring/api-endpoints/>
- Barzanallana, R. (2017). *Desarrollo de Aplicaciones WEB. Metaetiquetas en HTML*. Universidad de Murcia. <https://www.um.es/docencia/barzana/DAWEB/2017-18/daweb-tema-6-metaetiquetas-en-html.html>
- Calvo, D. (2018, abril 7). *Metodología XP Programación Extrema (Metodología ágil)*. Diego Calvo. <https://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/>
- Diego Navarro, R., & Cabrera Lozada, R. D. (2020). *Aplicación basada en arquitectura de microservicios* [Info:eu-repo/semantics/bachelorThesis, Universidad Complutense]. <https://eprints.ucm.es/id/eprint/62080/>
- Domínguez Dorado, M. (2016, octubre). *NetBeans IDE 4.1. La alternativa a Eclipse*. 32-34.
- Flores Cueto, J. J., & Bertolotti Zuñiga, C. (2015). *Diagrama de clases en UML*. <https://docplayer.es/13004382-Diagrama-de-clases-en-uml.html>
- García Carmona, J. (s. f.). *GRASP: Controlador*. GRASP. Recuperado 23 de noviembre de 2022, de <https://juan-garcia-carmona.blogspot.com/2012/09/grasp-controlador.html>
- García de León, A. (2002). Etapas en la creación de un sitio web. *Biblios*, 4(14). <https://www.redalyc.org/articulo.oa?id=16114408>
- Giraldo G, G. L., Acevedo, J. F., & Moreno N, D. A. (2011). Una ontología para la representación de conceptos de diseño de software. *Revista Avances en Sistemas e Informática*, 8(3). <https://www.redalyc.org/pdf/1331/133122679013.pdf>
- Guerra Soriano, P. M. G. (2021, junio 25). A cuatro años del Monitor de sitios web en Cuba. *Cubaperiodistas*. <https://www.cubaperiodistas.cu/index.php/2021/06/a-cuatro-anos-del-monitor-de-sitios-web-en-cuba/>
- Hamilton, D. (2020, abril 24). *Estándares de codificación de software y pautas de programación*. Parasoft. <https://es.parasoft.com/blog/an-ounce-of-prevention-software-safety-security-through-coding-standards/>
- Hernández León, R. A., & Coello González, S. (2012). *El proceso de investigación científica.: Vol. 2 Edición*. Editorial Universitaria (Cuba). <https://docplayer.es/108320563-El-proceso-de-investigacion-cientifica-2-edicion-rolando-alfredo-hernandez-leon-y-sayda-coello-gonzalez.html>

- Herrera, M. (2009). Ingeniería del software: Metodologías y ciclos de vida. *Laboratorio Nacional de Calidad Del Software*.  
[https://www.academia.edu/9795641/INGENIER%C3%8DA\\_DEL\\_SOFTWARE\\_METODOL OG%C3%8DAS\\_Y\\_CICLOS\\_DE\\_VIDA\\_Laboratorio\\_Nacional\\_de\\_Calidad\\_del\\_Software](https://www.academia.edu/9795641/INGENIER%C3%8DA_DEL_SOFTWARE_METODOL OG%C3%8DAS_Y_CICLOS_DE_VIDA_Laboratorio_Nacional_de_Calidad_del_Software)
- Lapiente, M. J. L. (2013). *Hipertexto, el nuevo concepto de documento en la cultura de la imagen*. [Tesis, Universidad Complutense].  
[http://www.hipertexto.info/documentos/tipos\\_buscador.htm](http://www.hipertexto.info/documentos/tipos_buscador.htm)
- Larman, C. (2003). *UML y patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado* (2da Edición). Pearson Educación.
- Llamas, J. (2021, agosto 7). *Proceso informático—Qué es, definición y concepto*. Encomedia.com. <https://economipedia.com/definiciones/proceso-informatico.html>
- Marcotte, E. (2010, mayo 25). *Responsive Web Design*. A List Apart. <https://alistapart.com/article/responsive-web-design/>
- Meléndez Valladares, S. M., Gaitan, M. E., & Pérez Reyes, N. N. (2016). *Metodología ágil de desarrollo de software Programación Extrema*. Universidad Nacional Autónoma de Nicaragua.
- Mousinho, A. (2020, junio 3). *Qué es SEO: La guía completa para que alcances la cima de Google*. Rock Content - ES. <https://rockcontent.com/es/blog/que-es-seo/>
- Oracle. (2018, diciembre 31). *¿Qué es java y para que la necesito?*  
[https://www.java.com/es/download/help/whatis\\_java.html](https://www.java.com/es/download/help/whatis_java.html)
- Pérez Porto, J. (2010, abril 6). *Definición de URL - Qué es, Significado y Concepto*. Definición.de. <https://definicion.de/url/>
- Pressman, R. (2010). *Ingeniería del Software. Un Enfoque Practico 7ma edición* (7ma Edición). McGraw Gill.  
[https://www.academia.edu/24308956/Ingenieria\\_del\\_Software\\_Un\\_Enfoque\\_Practico\\_7ma\\_edici%C3%B3n](https://www.academia.edu/24308956/Ingenieria_del_Software_Un_Enfoque_Practico_7ma_edici%C3%B3n)
- Pruebas de integración*. (2013). Portal Universidad de Sevilla. <https://www.us.es/estudiar/que-estudiar/oferta-de-grados/grado-en-ingenieria-aeroespacial/1970055>
- Raúl, E. (2009, junio 30). *Técnicas de Investigación de Campo: Método Análítico Sintético*. Técnicas de Investigación de Campo. <http://niveldostic.blogspot.com/2009/06/metodo-analitico-sintetico.html>
- Redirect Checker | Check your Statuscode 301 vs 302*. (s. f.). Recuperado 15 de noviembre de 2022, de <https://www.redirect-checker.org/>

- Sarmiento, J. (2013, abril 12). *Visión General de los Diagramas de Despliegue. UML*. <http://umldiagramadespliegue.blogspot.com/>
- Schwarz, T., & Müller, C. (2022). *Redirects & SEO - The Complete Guide*. Audisto. <https://audisto.com/guides/redirects/>
- Simons, M. (2020). *Infografik Spring Boot. Jax London*. [https://jaxlondon.com/wp-content/uploads/2020/08/Infografik\\_Spring\\_Boot\\_58651\\_ENG\\_v1-1.pdf](https://jaxlondon.com/wp-content/uploads/2020/08/Infografik_Spring_Boot_58651_ENG_v1-1.pdf)
- Sommerville, I. (2005). *Ingeniería del software (7ma Edición)*. Pearson Educación.
- Tiny URL Expander | Redirect Checker—WhereGoes*. (2022, mayo 23). <https://wheregoes.com/>
- URL Redirection Checker*. (s. f.). Geekflare Tools. Recuperado 15 de noviembre de 2022, de <https://geekflare.com/url-redirection-checker>
- Urosa Barreto, F. (2021). *Posicionamiento orgánico en buscadores (SEO): Estudio sobre el ranking de resultados en Google.es del sector educativo superior [Info:eu-repo/semantics/doctoralThesis, Universidad Complutense]*. <https://eprints.ucm.es/id/eprint/64419/>
- Vargas González, R. (2019). *Arquitectura de microservicio basada en contenedores*. <https://es.slideshare.net/ricardogonzalez19/20190427-arquitectura-de-microservicios-con-contenedores-143024230>
- Villegas Ballesta, M. (2015, mayo 27). *Arquitectura informacional. Aplicación de lógicas informacionales en las etapas iniciales del proyecto de arquitectura*. <https://doi.org/10.13140/RG.2.1.4149.0801>
- Website Monitoring, Website Monitoring Service, Server Monitoring: Site24x7*. (s. f.). Recuperado 15 de noviembre de 2022, de <https://www.site24x7.com/>