Identificación de requerimientos

Breve descripción:

El despliegue e implementación de sistemas, requiere de una serie de componentes de infraestructura y plataforma tecnológica, dentro de los cuales se encuentran los sistemas operativos, sistemas de gestión de base de datos, servidores web, lenguajes o intérpretes de programación, los cuales deben ser definidos antes del inicio del proyecto.

Abril 2024

Grupo de Ejecución de la Formación Virtual

Tabla de contenido

Introducción

En este componente formativo se estudiarán las generalidades que soportan la importancia de la realización de un alistamiento de infraestructura, para el adelanto de actividades en el marco del desarrollo y despliegue de "software", de acuerdo con el análisis de las necesidades particulares de los clientes.

Además, se presentan los diferentes tipos de configuraciones tradicionales de plataformas para el desarrollo e implementación de aplicaciones y servicios, como:

- LAMP
- WAMP
- XAMP
- WXCF
- XATMJ
- WIMA

Se trabajará en las generalidades de los requerimientos y sobre todo, situándolos en aquellos que no son funcionales en el proceso de desarrollo; posteriormente, se abordará la arquitectura y el despliegue de aplicaciones y servicios, que posibilitará conocer las necesidades de los clientes a partir de los requisitos no funcionales definidos.

Bienvenido.

1. Generalidades sobre requerimientos

El proceso de levantamiento de requerimientos es fundamental al momento de iniciar un proyecto de desarrollo de "software"; este permite identificar las características o funcionalidades que debe tener el "software" a construir o en algunos casos, también permite definir cuáles y qué tipo de restricciones deben ser cubiertas por el equipo de desarrollo, para que el producto sea recibido de manera satisfactoria por el cliente.

Este proceso se hace con el fin que el equipo de trabajo pueda entender qué tipo de funcionalidades, complejidad, tiempos, herramientas y tecnologías podrían ser empleadas durante el proceso de construcción y posterior implementación de la solución. Este es un proceso bastante crítico, porque si no se hace de manera correcta y exhaustiva, el desarrollo del proyecto puede no ser exitoso.

Según Sommerville (2011), los requerimientos se pueden clasificar en:

- **Funcionales**

  Hace referencia a lo que espera el usuario que ofrezca el sistema a nivel de servicios, como, por ejemplo: registrar comentarios, hacer búsquedas por nombre, etc.

- **No funcionales**

  Son las características y restricciones generales que debe satisfacer el sistema para poder ofrecer los servicios establecidos en los requisitos funcionales, con ciertos niveles de calidad requeridos, por ejemplo: arquitectura del sistema, tiempos de respuesta, capacidad de carga, etc.

Para el desarrollo de este componente y dada la naturaleza de los requerimientos, se abordarán los no funcionales, de tal manera que se pueda, a partir de la definición de la arquitectura, ser utilizada en el proceso de desarrollo de "software", suplir y dar cumplimiento con los requerimientos.

1.1. Requerimientos no funcionales

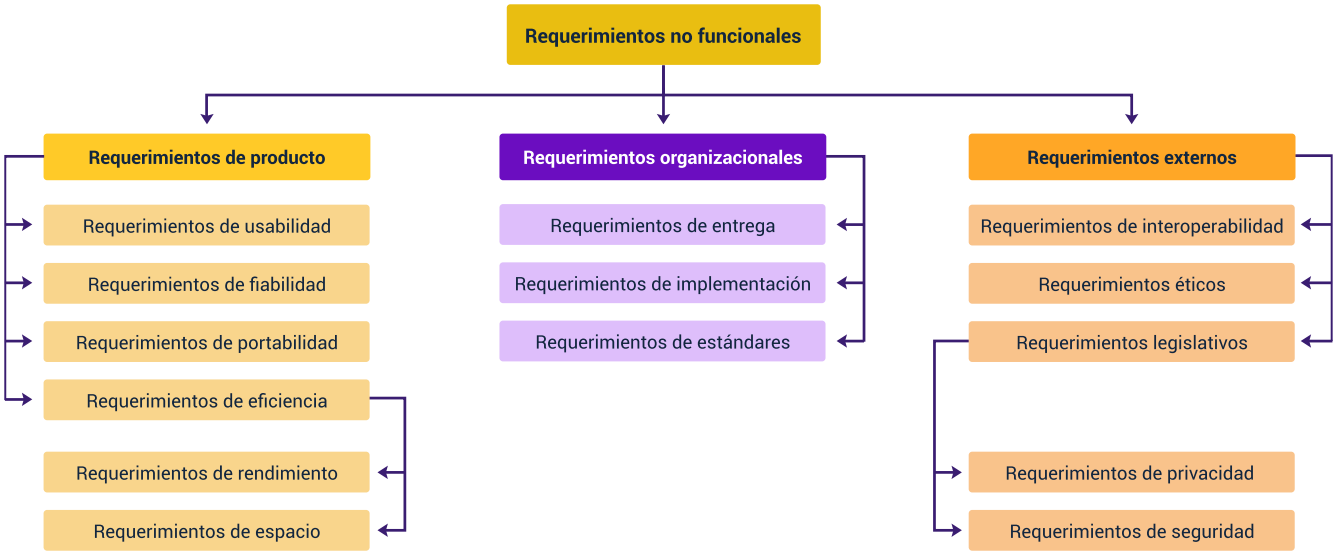
Estos no están relacionados de forma directa a funciones específicas del sistema, por esta razón algunos equipos de trabajo no le dedican el tiempo necesario para su definición.

Sin embargo, son los requerimientos no funcionales los que se asocian a propiedades clave que determinan en buena parte la calidad de los productos y servicios desarrollados, como, por ejemplo: la fiabilidad, los tiempos de respuesta a operaciones, la capacidad de almacenamiento, entre otros.

Los requerimientos no funcionales deben surgir de las necesidades del usuario y el problema particular que se intenta resolver, e involucran temas presupuestales, políticas de la organización, posible interoperabilidad con otros sistemas de "software" o "hardware", además diferentes tipos de políticas como: la privacidad, la seguridad, políticas ambientales, de manejo de datos, entre otras; por lo anterior algunas veces resolver este tipo de requerimientos puede ser más crítico que los mismos requerimientos funcionales. (Sommerville, 2011).

En la siguiente figura se presentan los diferentes tipos de requerimientos no funcionales, según Sommerville.

Figura 1. Clasificación de requerimientos no funcionales



¡Importante!

Los requerimientos de producto incluyen los requerimientos de eficiencia, de usabilidad, de fiabilidad y de portabilidad. Un ejemplo de un requerimiento no funcional de producto podría ser que el producto sea desarrollado en HTML5, sin el applet de Java, ni Ajax, esto con el ánimo de garantizar que sea compatible con la mayoría de los navegadores en el mercado actual.

Los requerimientos organizacionales provienen directamente de las características de la organización, objetivo para el producto o servicio de "software" a desarrollar, como, por ejemplo:

a) Estándares en los procesos.
b) Requerimientos de implementación, como lenguajes específicos de programación.

c) Metodologías de diseño.

d) Tiempos de entrega.

e) Documentación requerida por la organización.

En los requerimientos externos se clasifican todos aquellos requerimientos no funcionales adicionales que no corresponden específicamente al producto, su proceso de desarrollo o a la organización.

Entre este tipo de requerimientos están los relacionados con interoperabilidad con otros sistemas "software" o "hardware" de la empresa o terceros, lo relacionado con políticas y normativas gubernamentales o leyes, y los de tipo ético que se relacionan directamente con el público objetivo.

Un ejemplo de este tipo de requerimientos puede ser que debido a la política de protección de datos no se deben mostrar los datos personales de ninguna persona en los informes generados o por ejemplo, que el sistema sea compatible con los servicios de búsqueda de la Registraduría Nacional para el proceso de verificación de documentos en el registro.

¡Nota!

Se recomienda especificar los requerimientos no funcionales de forma cuantitativa o al menos que exista la definición de algunos criterios de aceptación, que facilite la verificación de su cumplimiento. Normalmente, los requerimientos no funcionales se especifican en el mismo documento de requerimientos, pero en una sección diferente a los requerimientos funcionales.

Existen dos corrientes para la documentación de requerimientos, independientemente de la corriente. Se recomienda utilizar la estructura de la siguiente

tabla, para la documentación de requerimientos no funcionales. Sin embargo, se pueden realizar los ajustes y modificaciones que considere enriquecen el proceso.

Tabla 1. Propuesta para especificación de requisitos no funcionales

Nombre del requerimiento	Componente
Identificador	[Identificador del requerimiento]
Descripción	[Describa la solicitud realizada por el cliente]
Propósito	[Escriba la finalidad del requerimiento]
Forma de verificación	[Escriba como se puede realizar la verificación de que el requerimiento fue alcanzado con éxito, en lo posible utilizar medidas cuantitativas]

2. Arquitectura de despliegue de aplicaciones y servicios

En el desarrollo de aplicaciones, es necesario tener claridad sobre la pertinencia de cada herramienta de desarrollo según el proyecto, debido a que el uso o no de una herramienta, tiene costos asociados que pueden impactar el alcance y recursos del proyecto.

Es importante analizar los diferentes tipos de licencias pagadas y libres a nivel de sistemas operativos, lo mismo aplica con herramientas usadas para el desarrollo de un producto como IDE, sistemas gestores de bases de datos, herramientas para la ejecución de pruebas, entre otros.

También se debe hacer un balance y un análisis serio sobre las mejores opciones, de acuerdo con las necesidades de los clientes y teniendo en cuenta los requisitos no funcionales definidos.

Por esta razón, es importante elegir las herramientas que harán parte de la arquitectura tecnológica definida para el desarrollo y posterior despliegue e implementación de la solución tecnológica, construida de acuerdo con el tipo de proyecto y tipo de cliente para el cual se pretende desarrollar el proyecto.

En el siguiente video, podemos estudiar los componentes de una arquitectura de “software”:

Video 1. Componentes de una arquitectura de “software”

Enlace de reproducción del video

Síntesis del video: Componentes de una arquitectura de “software”
Toda arquitectura definida en el proceso de construcción de un “software” debe incluir una serie de elementos o componentes que en su conjunto determinan la estructura o línea de trabajo entre todos los miembros del equipo de desarrollo. Para el inicio de un proyecto, la definición de los componentes dependerá de lo que se va a construir, por ejemplo, no es lo mismo desarrollar una aplicación web que una aplicación móvil. Para realizar lo anterior, se debe tener en cuenta:

Clientes y servidores: examinar qué servicios son requeridos en el servidor web: DNS, FTP, entre otros y de igual manera las aplicaciones de tipo cliente que son necesarias para acceder a estos servicios.

- Bases de datos.
- Sistemas Operativos.
- Lenguajes de programación.
- Niveles en sistemas jerárquicos.

En términos generales la arquitectura de desarrollo define los componentes del "software", su función e interacción con otros componentes de "software" y "hardware".

La selección de los elementos es lo que nos lleva a construir una plataforma de desarrollo estable, brindando un entorno común sobre la cual el equipo de creación realizará las actividades necesarias para la construcción del producto.

Adicionalmente, la plataforma es la base para la integración de los elementos del "software" y "hardware" que permiten el funcionamiento del sistema en entornos productivos una vez que se realizan los respectivos procesos de despliegue por lo cual es importante que la arquitectura de éste sea definida desde el inicio del proyecto.

2.1. Plataformas de desarrollo e implantación de aplicaciones y servicios

A continuación, se relacionan algunas de las plataformas de desarrollo e implantación de aplicaciones y servicios más comunes o de mayor popularidad, dependiendo del proyecto a construir:

LAMP

Es quizá la plataforma tecnológica más utilizada y popular en el ámbito del desarrollo web, es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Linux, el sistema operativo.
- Apache, el servidor web.
- MySQL/MariaDB, el gestor de bases de datos.
- PHP, PERL, PYTHON, el lenguaje de programación.

Este tipo de plataforma es muy común, debido a su fácil implementación y despliegue, además de utilizar herramientas tecnológicas todas de tipo GPL.

WAMP

Este tipo de plataformas se utilizan más en entornos locales (ON PREMISE) de despliegue, donde el acceso es a través de la red local al igual que la plataforma anterior utiliza casi los mismos componentes, a excepción del sistema operativo que en este caso es reemplazado por Windows en lugar de Linux. Incluye los siguientes componentes:

- Windows, el sistema operativo.
- Apache, el servidor web.
- MySQL/MariaDB, el gestor de bases de datos.
- PHP, PERL, PYTHON, el lenguaje de programación.

XAMPP

Es una plataforma tecnológica utilizada como un paquete integrando varios componentes de tipo multiplataforma, de ahí el origen de la primera letra X que

significa cualquiera de los diferentes sistemas operativos; este paquete es distribuido con licencia GNU y es una plataforma para despliegue de aplicaciones y servicios web de muy fácil instalación y utilización. Incluye los siguientes componentes:

- X, diferentes sistemas operativos (multiplataforma).
- Apache, el servidor web.
- MariaDB, el gestor de bases de datos.
- PHP, PERL, el lenguaje de programación.

WXCF

Esta plataforma utiliza ColdFusion para el desarrollo y despliegue de aplicaciones de tipo web, permite programar páginas o aplicaciones web a través de etiquetas similares a HTML y que además es multiplataforma, utiliza los siguientes componentes:

- Windows, el sistema operativo.
- X: puede utilizar servidores web como Apache o Internet Information services.
- Soporta bases de datos como: Sybase, Oracle, MySQL, SQL Server.

XATMJ

Este tipo de plataforma web es muy poco común encontrarla en los proveedores de hosting, se utiliza específicamente cuando el desarrollo web es realizado con JSP, incluye los siguientes componentes:

- X, diferentes sistemas operativos (multiplataforma).
- Apache.
- Tomcat como servidor web.
- MySQL, como gestor de bases de datos.

- JSP como lenguaje de programación.

WIMA

Este tipo de plataforma es muy popular en los desarrolladores que utilizan herramientas y plataformas tecnológicas propietarias de Microsoft, involucran desarrollos en ASP y ASP.net, e incluye los siguientes componentes:

- Windows, el sistema operativo.
- IIS ("Internet Information Services") como servidor web.
- MS SQL Server como gestor de bases de datos.
- ASP.net como lenguaje de programación.

2.2. Introducción a las máquinas virtuales y contenedores

En la definición de la arquitectura de despliegue de aplicaciones y servicios, es muy importante la definición de los componentes desde el punto de vista de la plataforma tecnológica que serán utilizados en el momento de iniciar el proceso de desarrollo y en el de hacer el despliegue o puesta en producción, para lo cual el equipo de desarrollo deberá elegir qué tipo de plataforma tecnológica será la utilizada.

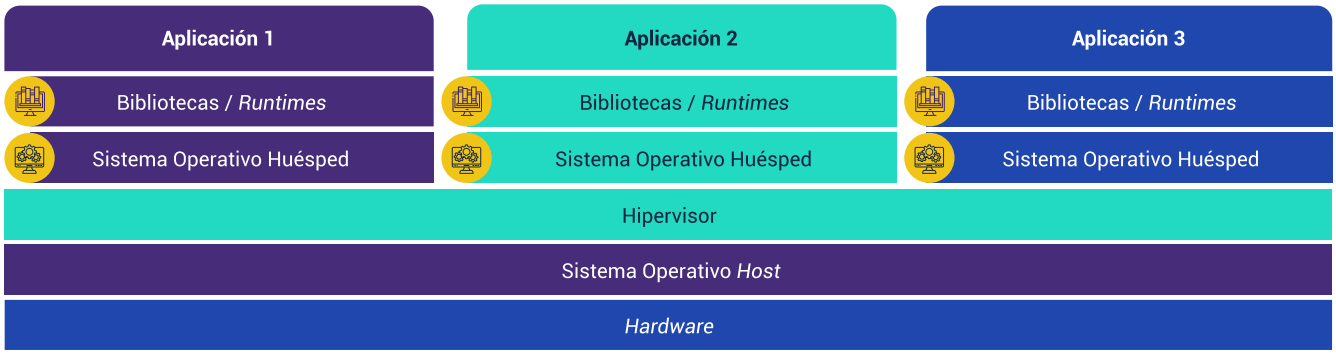
Si bien la plataforma tecnológica exige la selección de varios componentes como sistema operativo, servidor web, gestor de base de datos y lenguaje de programación, en la actualidad se debe tener claro qué tipo de despliegue se requiere para la construcción local del producto en las máquinas de los equipos de desarrollo y para el despliegue en producción (generalmente debe ser el mismo).

Lo anterior puede sugerir la necesidad de máquinas virtuales o contenedores, para lo cual se dará una definición que permita diferenciar un concepto de otro.

Máquinas virtuales

Las máquinas virtuales son un sistema operativo completo funcionando de manera aislada dentro de otro sistema operativo anfitrión (Alarcón, 2018). De igual manera, debe existir un componente de infraestructura de tipo "hardware" que soporte todo lo anterior, que en últimas es el servidor o equipo donde se haga el desarrollo y creación de la máquina virtual o si se trata del despliegue y puesta en producción final de la aplicación, puede referirse a un proveedor que suministra el "hardware" necesario que soportará la máquina virtual como Azure, Google Cloud, AWS, Digital Ocean, entre otras.

Figura 2. Funcionamiento de una máquina virtual



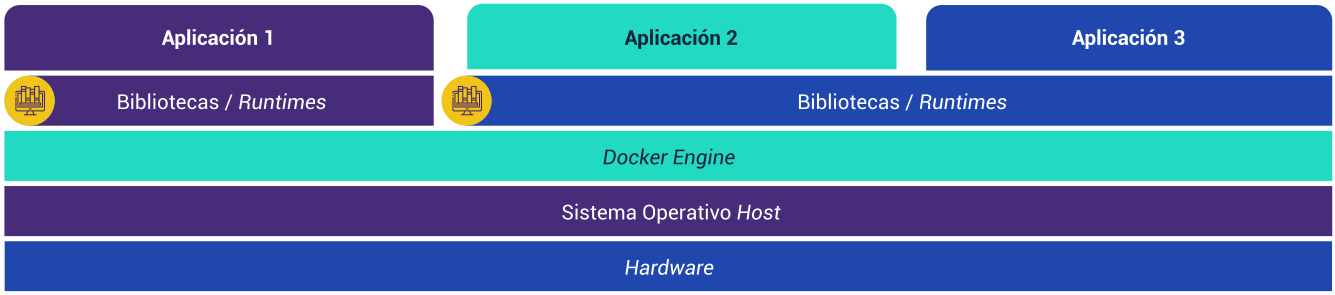
Contenedores

Los contenedores por su parte, a diferencia de las máquinas virtuales, no requieren tener un sistema operativo completo aislado, lo que hacen es aislar las aplicaciones, compartiendo los recursos del sistema operativo sobre el cual se ejecutan.

La contenerización de aplicaciones lo que hace es compartir todos esos recursos del equipo anfitrión entre todas las aplicaciones que se generen a través de contenedores, es decir, que ya no se requiere disponer de una capacidad determinada

de recursos como memoria RAM, procesador, almacenamiento y networking, como se hace en las máquinas virtuales, lo que permite optimizar el recurso.

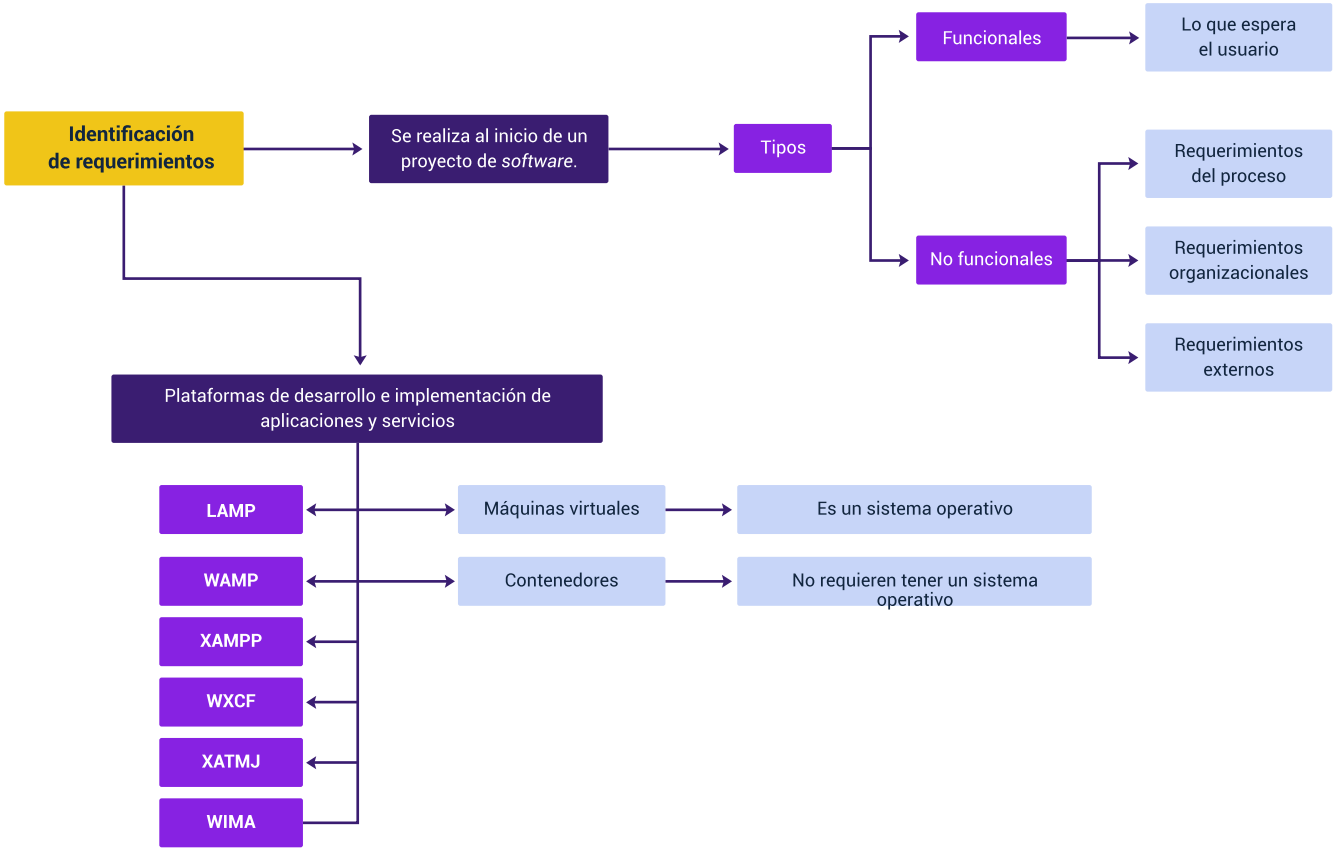
Figura 3. Funcionamiento contenedores



Al igual que en las máquinas virtuales, los contenedores requieren de un componente de infraestructura que soporte el sistema operativo anfitrión, el "Docker Engine" y la creación o aislamiento de las diferentes aplicaciones a través de contenedores que es entregado por el servidor de desarrollo, pruebas y puesta en producción, sea local o propio del equipo de desarrollo o entregado a través de un proveedor como Azure, Google Cloud, AWS, Digital Ocean, entre otras.

Síntesis

continuación, se muestra un mapa conceptual con los elementos más importantes desarrollados en este componente.



Material complementario

Tema	Referencia	Tipo de material	Enlace del recurso
Plataformas de desarrollo e implantación de aplicaciones y servicios.	Facultad Autodidacta. (2021). Instalar la pila de LAMP (Linux, apache, MariaDB y php) en Debian 10 [video]. YouTube	Video	https://youtu.be/AGF4fFuwCEU
Plataformas de desarrollo e implantación de aplicaciones y servicios.	Informática DP. (2021). WampServer 3.2.0 - Instalación paso a paso [video]. YouTube	Video	https://youtu.be/LIubtGr63RM
Plataformas de desarrollo e implantación de aplicaciones y servicios.	Ada Lovecode- Didacticode. (2021). Cómo descargar e instalar XAMPP en Windows 10 2020 para trabajar con Apache, PHP, MySQL, Perl [video]. YouTube	Video	https://youtu.be/DOZPG4V6-JU

Glosario

"**Hosting**": servicio de alojamiento de sitios web indispensable para que sea visualizado por cualquier usuario por medio de internet.

HTML: acrónimo en inglés de "HyperText Markup Language" (Lenguaje de Marcado de Hipertexto). HTML es el lenguaje de marcado estándar para crear páginas web y se usa para describir la estructura de una página web.

HTML5: es una versión de HTML.

IDE: acrónimo en inglés de "Integrated Development Environment" (Entornos de Desarrollo Integrado). Es un "software" especializado que ofrece servicios que facilitan el proceso de construcción de "software" a desarrolladores.

Interoperabilidad: capacidad de un sistema de comunicarse con otros sistemas para intercambiar y usar información entre ellos.

Referencias bibliográficas

Alarcón, J. M. (2018, 14 junio). ¿Qué diferencia hay entre Docker (Contenedores) y Máquinas virtuales (VMware, VirtualBox)? campusMVP.es. https://www.campusmvp.es/recursos/post/que-diferencia-hay-entre-docker-contenedores-y-maquinas-virtuales.aspx

Arroyave, M. H. T., & Cardona, D. (2012). Criterios de evaluación de plataformas de desarrollo de aplicaciones empresariales para ambientes web (Doctoral dissertation, Universidad Tecnológica de Pereira. Facultad de Ingenierías Eléctrica, Electrónica, Física y Ciencias de la Computación. Ingeniería de Sistemas y Computación).

Lazarte Méndez, J. P. (2019). Contenedores Docker como estrategia de virtualización (Doctoral dissertation).

Mariño, S. I., Godoy, M. V., Alfonzo, P. L., Acevedo, J. J., Solís, L. G., & Vázquez, A. F. (2012). Accesibilidad en la definición de requerimientos no funcionales. Revisión de herramientas. Multiciencias, 12(3), 305-312.

Pardo, M. R. V., Tapia, J. A. H., Moreno, A. S. G., & Sánchez, L. F. V. (2018). Comparación de tendencias tecnológicas en aplicaciones web.

Rodríguez Ramírez, A. M., & Obando Ortiz, F. R. (2005). Análisis comparativo de sistemas operativos de red (Bachelor`s thesis, QUITO/PUCE/2005).

Rojo, S. D. V. (2012). Requerimientos No funcionales para aplicaciones Web (Doctoral dissertation, Universidad Nacional de La Plata).

Sommerville I. (2011). Ingeniería del "software". México: Addison-Wesley

Torrico Rojas, L. (2019). Comparar “Máquinas virtuales vs Docker” (Doctoral dissertation).

Créditos

Nombre	Cargo	Centro de Formación y Regional
Milady Tatiana Villamil Castellanos	Líder del Ecosistema	Dirección General
Olga Constanza Bermudez Jaimes	Responsable de Línea de Producción	Centro de Servicios de Salud - Regional Antioquia
Jonathan Guerrero Astaiza	Experto temático	Centro de Teleinformática y Producción Industrial - Regional Cauca
Ana Catalina Córdoba Sus	Evaluadora Instruccional	Centro de Servicios de Salud - Regional Antioquia
Yerson Fabián Zárate Saavedra	Diseñador de Contenidos Digitales	Centro de Servicios de Salud - Regional Antioquia
Edward Leonardo Pico Cabra	Desarrollador Fullstack	Centro de Servicios de Salud - Regional Antioquia
Edgar Mauricio Cortés García	Actividad Didáctica	Centro de Servicios de Salud - Regional Antioquia
Daniela Muñoz Bedoya	Animador y Productor Multimedia	Centro de Servicios de Salud - Regional Antioquia
Jaime Hernán Tejada Llano	Validador de Recursos Educativos Digitales	Centro de Servicios de Salud - Regional Antioquia
Margarita Marcela Medrano Gómez	Evaluador para Contenidos Inclusivos y Accesibles	Centro de Servicios de Salud - Regional Antioquia
Daniel Ricardo Mutis Gómez	Evaluador para contenidos inclusivos y accesibles	Centro de Servicios de Salud - Regional Antioquia