INSTITUTO POLITÉCNICO NACIONAL

# UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA Y TECNOLOGIAS AVANZADAS

INGENIERIA TELEMATICA

INGENERIA WEB

# COSTUMERS & WORKERS

JOSUE ANTONIO JIMENEZ ASTORGA

JUAN CARLOS RAMIREZ OROZCO

GRUPO: 2TM3

PROFESOR: FRANCISCO ANTONIO POLANCO MONTELONGO

## FECHA: 21 DE JUNIO DE 2024

**INDICE**

Portada …………………………………….……………………………………………… 1

índice ...…………………………………….……………………………………………… 2

Análisis de requerimientos ………………..………………………………………………. 3

Requerimientos funcionales ……..…………………………………...….………... 3

Requerimientos NO funcionales ..……………………………………..…….…… 5

Diseño ……………………………………………………………………………..…….... 5

Arquitectura del sistema …..…………………………………………..…….……. 9

Modelado UML ……….….……………………………………………....………. 11

Maquetado y diagrama de navegación ……….…………………………..………. 12

Modelo de datos ………….………………………………………………………. 13

Referencias ………………………………………………………………………….……. 14

**Análisis de requerimientos**

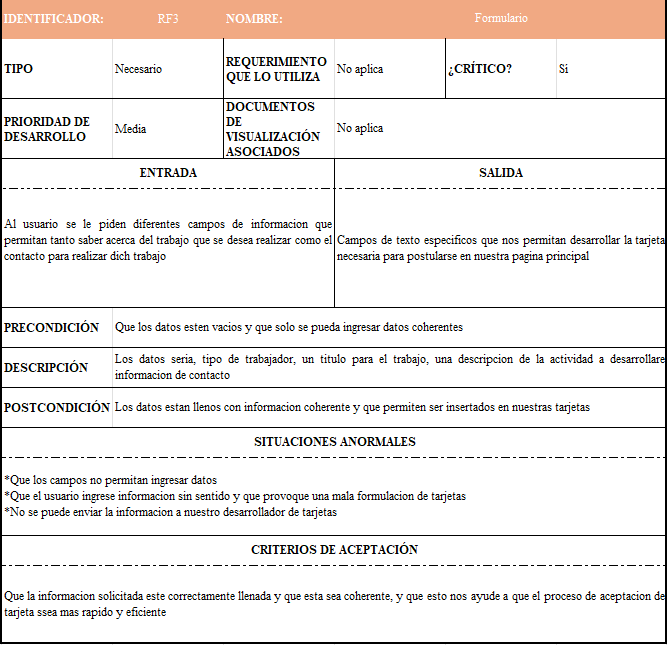
En el mundo del desarrollo de software y aplicaciones, los requerimientos son críticos. Son los que definen la funcionalidad y el propósito de una pieza particular de software o aplicación. Sin estos bien definidos, es difícil crear algo que satisfaga las necesidades y expectativas del usuario. Pero primero, ¿qué significan los requerimientos en el desarrollo de software? **requerimientos de software** son simplemente una descripción de lo que un programa de software en particular debe hacer. Actúan como pautas para que los desarrolladores creen un producto funcional que satisfaga las necesidades de los usuarios. Siga leyendo para descubrir por qué los requerimientos son importantes para usted como ingeniero de software.

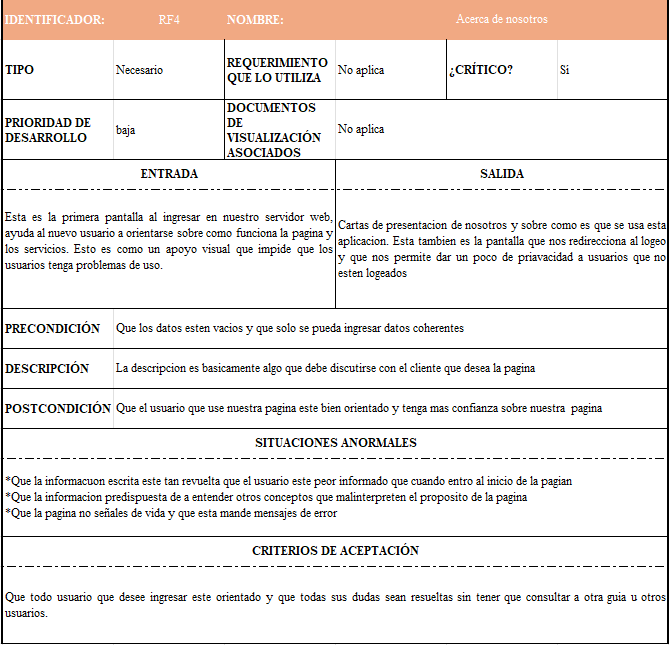
Generalmente hay dos tipos de requerimientos en el desarrollo de software y aplicaciones: **funcionales y no funcionales**. Los requerimientos funcionales especifican lo que debe hacer un sistema, mientras que los requerimientos no funcionales especifican cómo debe comportarse el sistema.

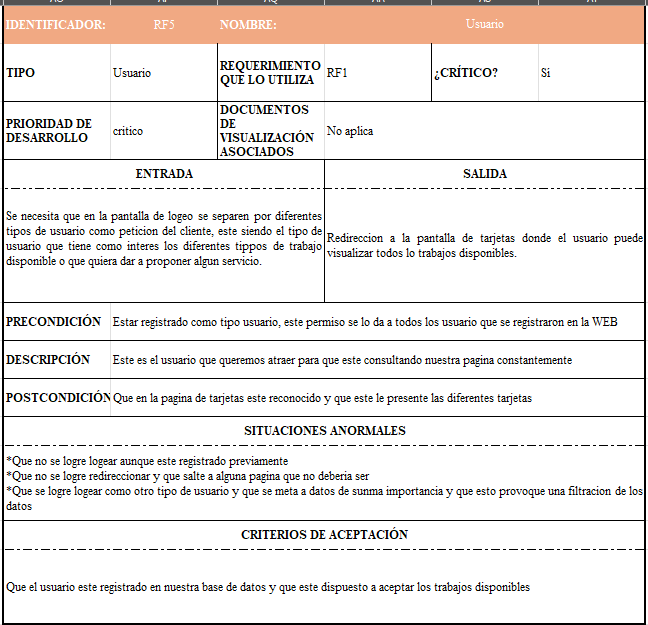
**Requerimientos funcionales**

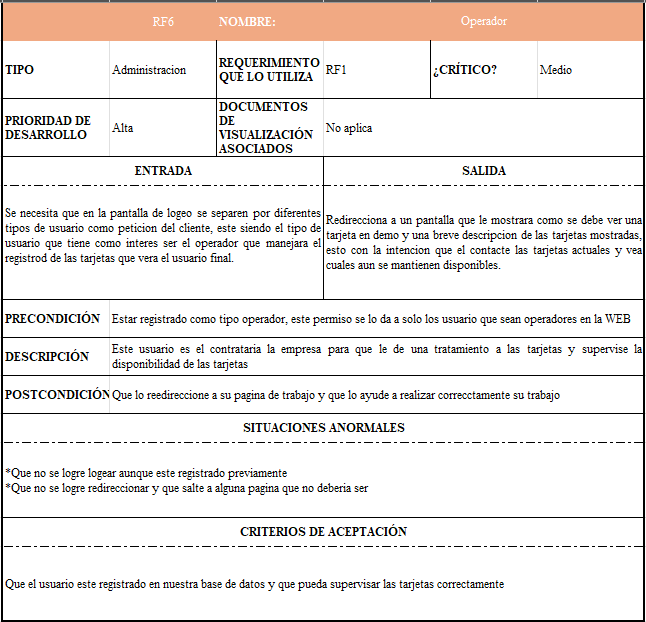


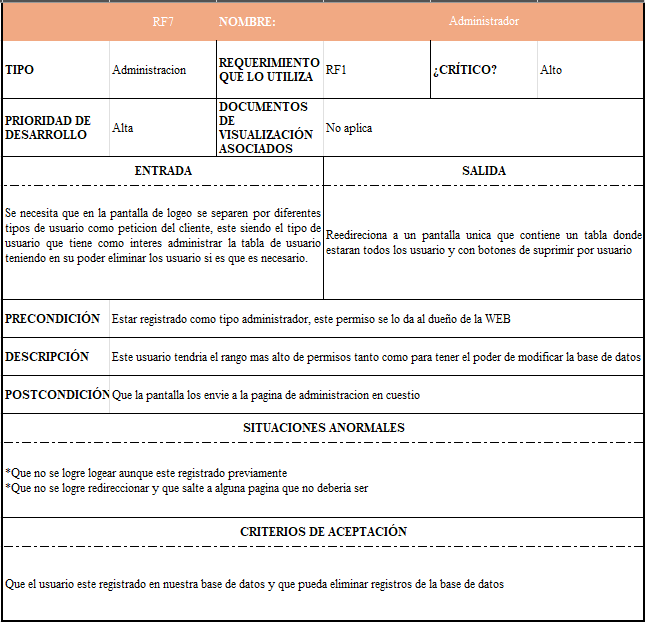
****

****

****

****

****

****

**Requerimientos NO funcionales**

El uso de hash en el registro de usuario proporciona varias ventajas clave en términos de seguridad. A continuación, se describen algunas de las principales:

1. \*Protección de Contraseñas\*:

- \*Irreversibilidad\*: Las funciones hash están diseñadas para ser irreversibles, lo que significa que, a partir del hash, no se puede obtener la contraseña original. Esto asegura que incluso si un atacante obtiene acceso a la base de datos de contraseñas, no podrá recuperar fácilmente las contraseñas en texto plano.

- \*Almacenamiento Seguro\*: En lugar de almacenar las contraseñas en texto claro, se almacena su hash. Así, incluso si la base de datos es comprometida, las contraseñas no estarán directamente accesibles.

2. \*Uso de Salting\*:

- \*Única Hash para Cada Usuario\*: Al añadir un valor aleatorio conocido como "salt" a cada contraseña antes de aplicar la función hash, se garantiza que incluso contraseñas idénticas tendrán hashes diferentes. Esto evita ataques de tablas de búsqueda (lookup tables) y rainbow tables.

- \*Resistencia a Ataques de Diccionario\*: Los ataques de diccionario buscan contraseñas comunes pre computadas con sus hashes. Con salting, cada hash es único, lo que hace que estos ataques sean imprácticos.

3. \*Funciones Hash Criptográficas\*:

- \*Funciones Dedicadas\*: Se utilizan funciones hash criptográficas como SHA-256, bcrypt, scrypt o Argon2, que están diseñadas específicamente para la seguridad. Estas funciones son resistentes a colisiones (dos entradas diferentes que producen el mismo hash) y están optimizadas para ser difíciles de computar en masa.

- \*Resistencia a Ataques de Fuerza Bruta\*: Funciones como bcrypt, scrypt y Argon2 están diseñadas para ser computacionalmente intensivas y ralentizar los intentos de fuerza bruta, limitando la tasa a la cual un atacante puede probar posibles contraseñas.

4. \*Integridad y Verificación\*:

- \*Verificación de Contraseñas\*: Durante el inicio de sesión, la contraseña ingresada por el usuario se combina con el mismo salt y se hashea. Si el hash resultante coincide con el almacenado en la base de datos, se puede verificar la autenticidad de la contraseña sin necesidad de almacenarla en texto claro.

- \*Detección de Alteraciones\*: Los hashes pueden ser utilizados para detectar si los datos han sido alterados, proporcionando un mecanismo para verificar la integridad de la información almacenada.

En resumen, el uso de hashes en el registro de usuarios proporciona una fuerte protección contra la divulgación de contraseñas, ataques de fuerza bruta, y otros intentos de comprometer la seguridad de las cuentas de los usuarios. Esto asegura que incluso si una base de datos es comprometida, las contraseñas de los usuarios permanezcan seguras y no sean fácilmente accesibles para los atacantes.

**Diseño**

**El diseño web implica trabajo relacionado con el layout y diseño de páginas online, así como la producción de contenido, aunque generalmente se aplica a la creación de sitios web.**

En este caso, los diseñadores web crean las páginas utilizando lenguajes de marcado como HTML o XML.

Por otro lado, la parte visual de los sitios está a cargo del CSS, término utilizado para estilizar los elementos escritos en HTML.

Por lo tanto, es común que los diseñadores web utilicen ambos para construir un site porque juntos definirán cómo aparecerán las páginas en los navegadores.

El proceso de creación de sitios se puede realizar desde herramientas como Adobe Dreamweaver (que requerirá que el profesional sepa más sobre los códigos) y CMS que son plataformas de administración de contenido.

WordPresses uno de los más conocidos y prácticos porque tiene una serie de templates que pueden basar el sitio. El web designer tiene la tarea de instalarlos y personalizarlos según lo que cada cliente quiera.

El [diseño de una página o sitio web](https://www.hubspot.es/products/cms/web-design?hubs_post=blog.hubspot.es/website/que-es-css&hubs_post-cta=dise%C3%B1o%20de%20una%20p%C3%A1gina%20o%C2%A0sitio%20web) necesita un lenguaje común para que los navegadores puedan interpretarlo y mostrarlo a las personas de la forma correcta. Pero también lo requieren para que otros diseñadores hagan las modificaciones necesarias para su optimización.

Uno de estos lenguajes es el CSS, un gran aliado del HTML que transforma la experiencia de tus visitantes. Con este artículo, aprenderás todo lo que necesitas saber sobre sus funciones y ventajas.

¿Qué es el CSS?

CSS son las siglas en inglés para «hojas de estilo en cascada» (Cascading Style Sheets). Básicamente, es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un usuario las visita. Funciona junto con el lenguaje HTML que se encarga del contenido básico de los sitios.

Se les denomina hojas de estilo «en cascada» porque puedes tener varias y una de ellas con las propiedades heredadas (o «en cascada») de otras.

Para muchas personas, una simple plantilla de blog es suficiente. Aun así, cuando quieras personalizar la apariencia de un sitio, necesitarás implementar CSS que, en conjunto con un buen CMS, te ayudará a potenciar el alcance de tu contenido.

¿Para qué sirve?

Con CSS, puedes crear reglas para decirle a tu sitio web cómo quieres mostrar la información y guardar los comandos para elementos de estilo (como fuentes, colores, tamaños, etc.) separados de los que configuran el contenido.

Además, puedes crear formatos específicos útiles para comunicar tus ideas y producir experiencias más agradables, en el aspecto visual, para los usuarios del sitio web.

* **Separación de la estructura y la presentación.** CSS permite separar el contenido HTML de su presentación visual. Es decir, te permite mantener el código HTML limpio y estructurado, mientras que el estilo se define en un archivo CSS separado. Este orden mejora la legibilidad del código y facilita el mantenimiento, así como la actualización de los estilos.
* **Consistencia y mantenibilidad.** Al utilizar CSS, puedes aplicar estilos de manera consistente a un sitio o aplicación web. Los estilos se definen una vez y se aplican a múltiples elementos en las páginas, lo que favorece la renovación de la apariencia visual de todo el proyecto.
* **Eficiencia en el rendimiento.** CSS permite cargar estilos externos en un archivo separado. Con esto, el navegador almacenará en caché los estilos y los aplicará a todas las páginas del sitio, lo que mejora el rendimiento al reducir la cantidad de datos que deben transferirse entre el servidor y el cliente.
* **Flexibilidad y control.** Asimismo, ofrece una amplia gama de propiedades y selectores con los que tendrás un control preciso sobre el estilo de los elementos HTML. Podrás modificar, de forma sencilla, los colores, fuentes, márgenes, tamaños, diseños, etc. Gracias a estas características, podrás personalizar y adaptar a varios dispositivos y tamaños de pantalla tu sitio web.

**Bootstrap es un framework CSS utilizado en aplicaciones front-end** — es decir, en la pantalla de interfaz con el usuario— para desarrollar aplicaciones que se adaptan a cualquier dispositivo.

En wordPress, por ejemplo, puede instalarse como tema o usarse para el desarrollo de plugins o, incluso, dentro de ellos para estilizar sus funciones. El propósito del framework es ofrecerle al usuario una experiencia más agradable cuando navega en un sitio.

Por esta razón, tiene varios recursos para configurar los estilos de los elementos de la página de una manera simple y eficiente, además de facilitar la construcción de páginas que, al mismo tiempo, están [adaptadas para la web y para dispositivos móviles](https://rockcontent.com/es/blog/diseno-responsivo/).

Lo anterior demuestra por qué es importante conocer una estructura potencial de este tipo.

¿Qué es bootstrap?

Bootstrap es un frameworks CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.

Inicialmente, se llamó Twitter Blueprint y, un poco más tarde, en 2011, se transformó en código abierto y su nombre cambió para Bootstrap. Desde entonces fue actualizado varias veces y ya se encuentra en la versión 4.4.

El framework combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

Además de todas las características que ofrece el framework, **su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles**.

Esto significa que las páginas están diseñadas para funcionar en desktop, tablets y smartphones, de una manera muy simple y organizada.

Bootstrap está constituido por una serie de archivos CSS y JavaScript responsables de asignar características específicas a los elementos de la página.

Hay un archivo principal llamado bootstrap.css, que contiene una definición para todos los estilos utilizados. Básicamente, la estructura del framework se compone de dos directorios:

* **css:** contiene los archivos necesarios para la estilización de los elementos y una alternativa al tema original;
* **js:** contiene la parte posterior del archivo bootstrap.js (original y minificado), responsable de la ejecución de aplicaciones de estilo que requieren manipulación interactiva.

Para asignarle una característica a un elemento, simplemente debemos informar la clase correspondiente en la propiedad “class” del elemento que será estilizado.

**Arquitectura del sistema**

El mundo se ha trasladado a Internet, y las aplicaciones web se han convertido en los nuevos lugares de trabajo y tiendas comerciales. Para dar cabida a la variedad de propósitos a los que sirven las aplicaciones web modernas, cada una de ellas tiene que estar diseñada para ofrecer un alto rendimiento y personalización.

Las arquitecturas de las aplicaciones web resuelven este problema.  
  
La arquitectura de las aplicaciones web define cómo se estructuran los distintos componentes de una aplicación basada en la web. Esta arquitectura es muy específica para la naturaleza y el propósito de la aplicación web. Elegir una arquitectura equivocada para tu aplicación web puede causar estragos en tu negocio.

En esta guía, desglosaremos el concepto de arquitectura de la aplicación web y comprenderemos cómo afecta a la experiencia del usuario final de tu aplicación.

La arquitectura de una aplicación web es, sin duda, una de las partes más importantes de tu aplicación web. Si eliges desarrollar tu aplicación web con una arquitectura específica en mente, seguro que obtendrás muchos beneficios a la hora de mantener y hacer crecer tu aplicación.

Sin embargo, la elección de la arquitectura adecuada amplía aún más estos beneficios.

**Arquitectura orientada a servicio**

La arquitectura orientada a servicios es una de las alternativas más populares a la forma tradicional de construir aplicaciones en forma de monolito. En esta arquitectura, las aplicaciones web se dividen en servicios que representan una unidad funcional de negocio cada uno. Estos servicios están débilmente acoplados e interactúan entre sí mediante el paso de mensajes.

La arquitectura orientada a servicios añade estabilidad y escalabilidad al stack tecnológico de tu aplicación. Sin embargo, el tamaño de los servicios en la SOA no está claramente definidos y suelen estar vinculados a los componentes de negocio, no a los componentes técnicos; de ahí que el mantenimiento pueda ser a veces un problema.

**Ventajas de arquitectura orientada a servicio**

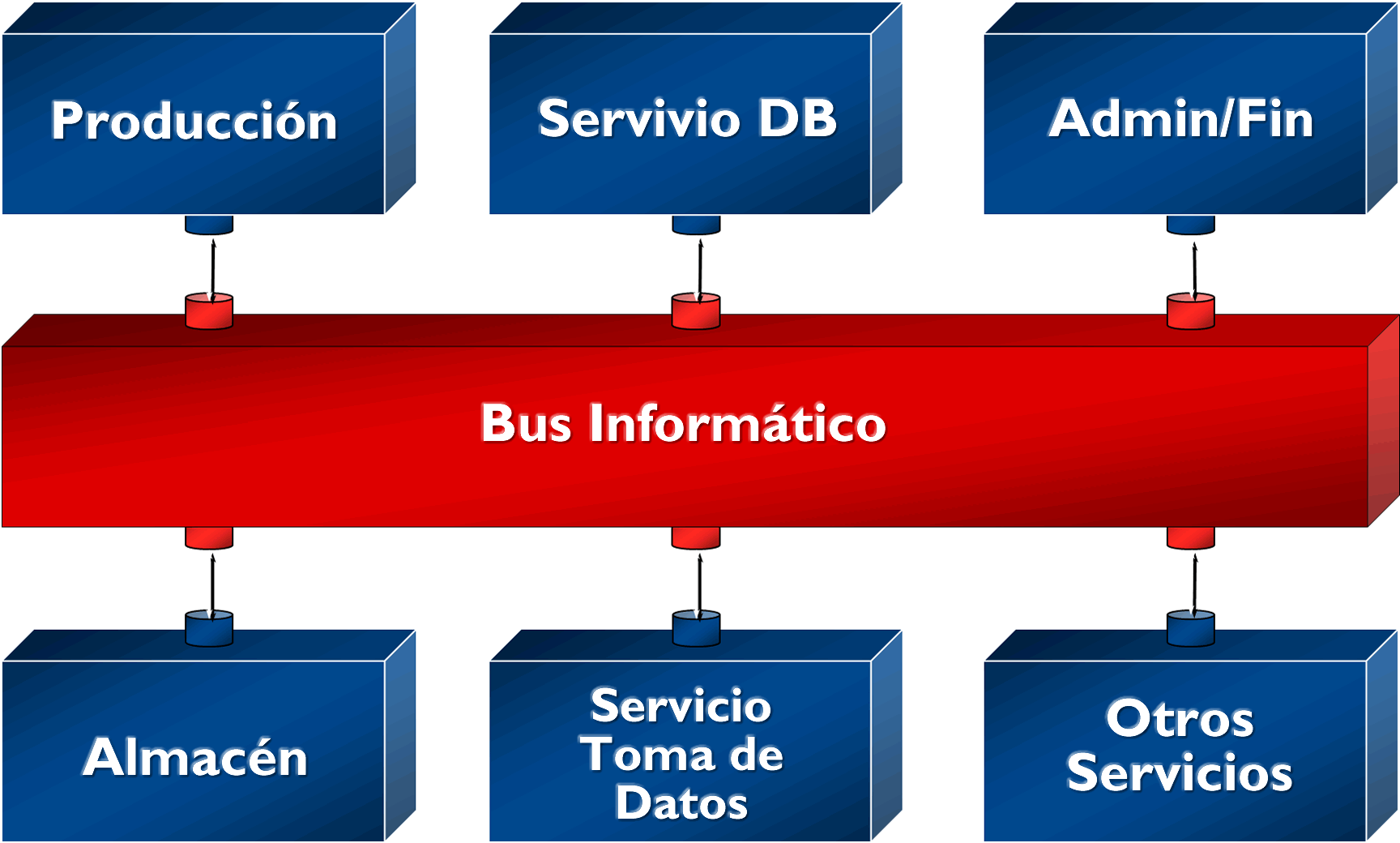
Las principales ventajas de la arquitectura orientada a servicios son:

* Esta arquitectura ayuda a construir aplicaciones altamente escalables y fiables.
* Los componentes son reutilizables y se comparten para mejorar los esfuerzos de desarrollo y mantenimiento.

**Contras de arquitectura orientada a servicio**

Los inconvenientes de la arquitectura orientada a servicio pueden ser:

* Las aplicaciones SOA aún no son 100% flexibles, ya que el tamaño y el alcance de cada servicio no son fijos. Puede haber servicios del tamaño de aplicaciones empresariales que pueden ser difíciles de mantener.
* La compartición de componentes introduce dependencias entre los servicios.



**Modelado UML**

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.

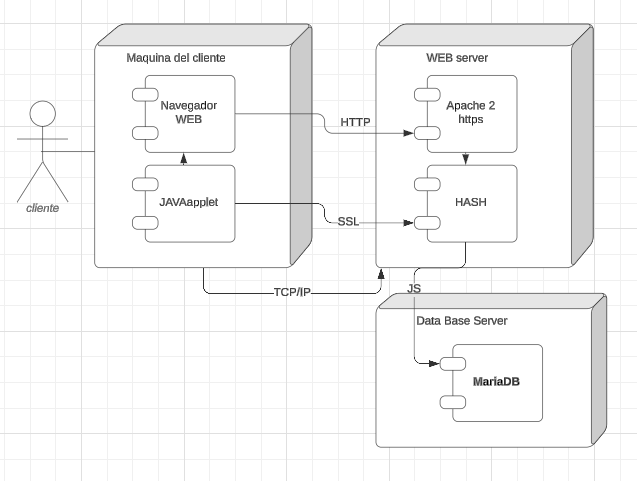
Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.

UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML. UML guarda una relación directa con el análisis y el diseño.

Diagrama de implementación: Ilustra el hardware del sistema y su software. Útil cuando se implementa una solución de software en múltiples máquinas con configuraciones únicas.

Un diagrama de implementación modela la implementación física y la estructura de los componentes de hardware. Los diagramas de implementación muestran dónde y cómo operarán los componentes de un sistema en conjunto con los demás.

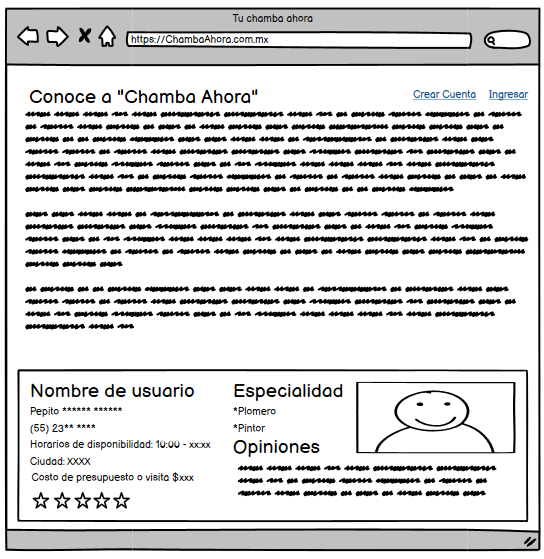
1. Al trazar un diagrama de implementación, usa la misma notación que usas para un diagrama de componentes.
2. Usa un cubo 3D para modelar un nodo (lo cual representa una máquina física o máquina virtual).
3. Etiqueta el nodo con el mismo estilo que se usa para los diagramas de secuencia. Agrega otros nodos según sea necesario, luego conéctalos con líneas.

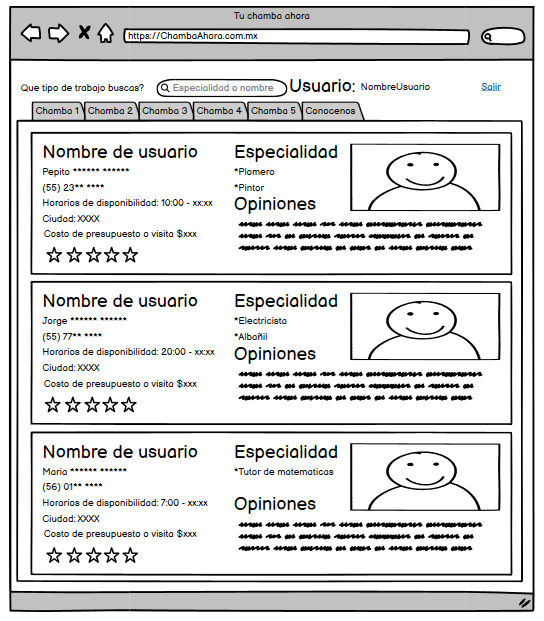


**Maquetado y diagrama de navegación**

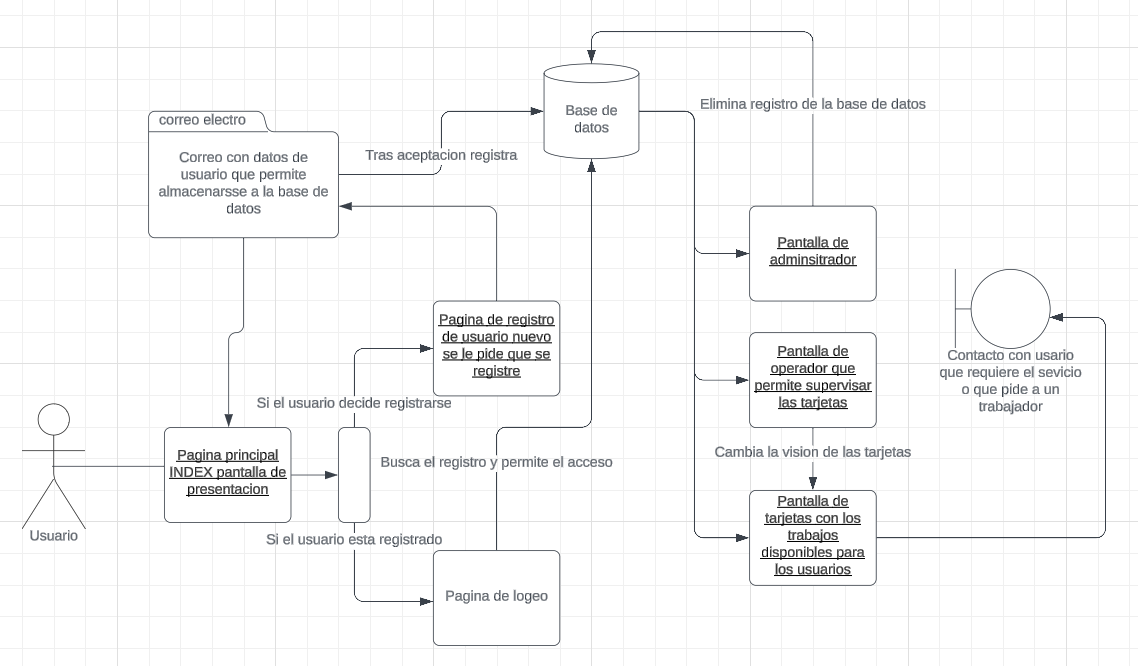
La maquetación web es el proceso de planificación, conceptualización y organización de contenido de una página web, es decir, asegura que los textos, las imágenes, los vídeos, como todo lo que quieras añadir a una página web, están estructurados para crear una buena experiencia de usuario.

Para comprender a cabalidad qué es la maquetación web, **es necesario entender que esta va más allá de la estética para enfocarse también en la funcionalidad general de una página web, teniendo en cuenta la arquitectura de la información.** Aunque también involucra el diseño de las aplicaciones web, móviles y el diseño de interfaz de usuario.

****

**** ****

Un mapa de navegación web es una representación visual de las páginas que conforman un sitio web y la información que contendrán. Permite hacer una jerarquía fácilmente comprensible y facilitar que los usuarios puedan acceder a cada contenido de manera intuitiva y ordenada.

****

**Modelo de datos**

Un modelo de datos integral y optimizado ayuda a crear una base de datos lógica simplificada que elimina la redundancia, reduce los requisitos de almacenamiento y permite una recuperación eficiente. También equipa a todos los sistemas con una "única fuente de verdad" –que es esencial para operaciones eficaces y cumplimiento regulatorio–. El modelado de datos es un paso clave en dos funciones vitales de una empresa digital.

|  |  |  |
| --- | --- | --- |
| Nombre de atributo | Tipo | Descripcion |
| ID | int | El numero identificador de la tabla |
| nombre | varchar(50) | Nombre del usuario a registrar |
| apellidos | varchar(50) | Apellido del usuario a registrar |
| usuario | varchar(50) | Nombre de ingreso a la pagina |
| email | varchar(100) | Correo de contacto del usuario |
| telefono | varchar(20) | Telefono de contacto del usuario |
| password | varchar(255) | Contraseña cifrada por el metodo HASH |
| fecha\_nacimiento | date | Fecha de nacimiento del usuario |
| direccion | text | Media direccion del usuario |
| tipo | varchar(20) | Este es el que define que tipo de logeo se le dará al momento de solicitar la entrada a la WEB |

**Referencias**

* Casteleyn, S., Daniel, F., & Dolog, P. (2010). *Engineering Web Applications*. Springer.
* Content, R. R. (2019, junio 21). *Aprende qué es el diseño web y lo que hace un profesional de esta área*. Rock Content - ES; Rock Content. https://rockcontent.com/es/blog/diseno-web/
* *¿Cuáles son algunos ejemplos de requisitos funcionales para un sitio web de comercio electrónico?* (2023, marzo 4). Linkedin.com; www.linkedin.com. https://es.linkedin.com/advice/1/what-some-examples-functional-requirements-e-commerce?lang=es
* de Octubre de, I. S. 830-1998 22. (s/f). *Especificaci ́on de Requisitos seg ́un el est ́andar de IEEE 830*. Ucm.es. Recuperado el 30 de abril de 2024, de https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf
* Dick, J., Hull, E., & Jackson, K. (2017). *Requirements Engineering* (4a ed.). Springer International Publishing.
* Flores, A. (2022, abril 17). *¿Qué es la maquetación web? ¡Enamora a tus usuarios a primera vista!* https://www.crehana.com. https://www.crehana.com/blog/transformacion-digital/que-es-maquetacion-web/
* Harsh, K. (2022, octubre 13). *¿Qué es la Arquitectura de las Aplicaciones Web? Desglosando una Aplicación Web*. Kinsta®; Kinsta. https://kinsta.com/es/blog/arquitectura-aplicaciones-web/
* Northware. (2022, mayo 26). *Requerimientos en el desarrollo de software y aplicaciones*. Northware. https://www.northware.mx/blog/requerimientos-en-el-desarrollo-de-software-y-aplicaciones/
* *password\_hash*. (s/f). Php.net. Recuperado el 8 de junio de 2024, de https://www.php.net/manual/es/function.password-hash.php
* *Qué es el lenguaje unificado de modelado (UML)*. (s/f). Lucidchart. Recuperado el 30 de mayo de 2024, de https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml
* *Requerimientos funcionales: Ejemplos*. (s/f). Pmoinformatica.com. Recuperado el 20 de abril de 2024, de https://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html
* Santos, D. (2023, julio 25). *Introducción al CSS: qué es, para qué sirve y otras 10 preguntas frecuentes*. Hubspot.es. https://blog.hubspot.es/website/que-es-css
* (S/f). Sap.com. Recuperado el 13 de mayo de 2024, de https://www.sap.com/latinamerica/products/technology-platform/datasphere/what-is-data-modeling.html