



SOFTWARE PROJECT LEADER

PROFESSIONAL CERTIFICATION



SPLPC™ Versión 032025



...

SOFTWARE PROJECT LEADER PROFESSIONAL CERTIFICATION

SPLPC®

SPLPC™ Versión 032025



¿Quién es Certiprof®?

Certiprof® es una entidad certificadora fundada en los Estados Unidos en 2015, ubicada actualmente en Sunrise, Florida.

Nuestra filosofía se basa en la creación de conocimiento en comunidad y para ello su red colaborativa está conformada por:

- **Nuestros Lifelong Learners (LLL)** se identifican como Aprendices Continuos, lo que demuestra su compromiso inquebrantable con el aprendizaje permanente, que es de vital importancia en el mundo digital en constante cambio y expansión de hoy. Independientemente de si ganan o no el examen.
- Las universidades, centros de formación, y facilitadores en todo el mundo forman parte de nuestra red de aliados **ATPs (Authorized Training Partners.)**
- **Los autores (co-creadores)** son expertos de la industria o practicantes que, con su conocimiento, desarrollan contenidos para la creación de nuevas certificaciones que respondan a las necesidades de la industria.
- **Personal Interno:** Nuestro equipo distribuido con operaciones en India, Brasil, Colombia y Estados Unidos está a cargo de superar obstáculos, encontrar soluciones y entregar resultados excepcionales.



Nuestras Afiliaciones

Memberships



Digital badges issued by



IT Certification Council – ITCC

Certiprof® es un miembro activo de ITCC.

Una de las ventajas de hacer parte del ITCC es como líderes del sector colaboran entre sí en un formato abierto para explorar maneras nuevas o diferentes formas de hacer negocios que inspiran y fomentan la innovación, estableciendo y compartiendo buenas prácticas que nos permiten extender ese conocimiento a nuestra comunidad.

Certiprof ha contribuido a la elaboración de documentos blancos en el Career Path Ways Taskforce, un grupo de trabajo que se implementó internamente para ofrecer a los estudiantes la oportunidad de saber qué camino tomar después de una certificación.

Algunos de los miembros del ITCC

- **IBM**
- **CISCO**
- **ADOBE**
- **AWS**
- **SAP**
- **GOOGLE**
- **ISACA**



Certiprof® es un miembro corporativo de Agile Alliance.

Al unirnos al programa corporativo Agile Alliance, continuamos empoderando a las personas ayudándolas a alcanzar su potencial a través de la educación. Cada día, brindamos más herramientas y recursos que permiten a nuestros socios formar profesionales que buscan mejorar su desarrollo profesional y sus habilidades.

<https://www.agilealliance.org/organizations/certiprof/>



Esta alianza permite que las personas y empresas certificadas con Certiprof® cuenten con una distinción a nivel mundial a través de un distintivo digital.

Credly es el emisor de insignias más importante del mundo y empresas líderes en tecnología como IBM, Microsoft, PMI, Nokia, la Universidad de Stanford, entre otras, emiten sus insignias con Credly.

Empresas que emiten insignias de validación de conocimiento con Credly:

- **IBM**
- **Microsoft**
- **PMI**
- **Universidad de Stanford**
- **Certiprof**



Insignias Digitales



- Según el estudio del IT Certification Council (ITCC), años atrás, la gente sabía muy poco sobre las insignias digitales. Hoy, grandes empresas e instituciones educativas de todo el mundo expiden insignias.
- Las insignias digitales contienen metadatos detallados sobre quién las ha obtenido, las competencias requeridas y la organización que las ha expedido. Algunas insignias incluso están vinculadas a las actividades necesarias para obtenerlas.
- Para las empresas e instituciones educativas, las insignias y la información que proporcionan son tan importantes que muchas decisiones, como las de contratación o admisión, se basan en los datos que aportan.

Insignias Digitales:
¿Qué Son?



¿Por qué son importantes?



- **Facilidad de Compartir y Verificar Logros:**

Las insignias digitales permiten a los profesionales mostrar y verificar sus logros de manera instantánea y global. Según un informe de Credly, **los perfiles de LinkedIn con insignias digitales reciben un 40% más de atención por parte de reclutadores y empleadores.**

- **Visibilidad en Plataformas Digitales:**

En una encuesta realizada por Pearson y Credly, el **85%** de los usuarios que obtuvieron insignias digitales **las compartieron en LinkedIn**, y el **75%** reportó que esto mejoró su **credibilidad profesional en sus redes**. Además, el **76%** de los empleadores encuestados afirmó que las insignias digitales les ayudan a identificar rápidamente habilidades específicas.



¿Por qué son importantes?

- **Impacto en la Contratación:**

Un estudio de la **Asociación Internacional de Gestión de Proyectos (PMI)** encontró que los candidatos que muestran insignias digitales de gestión de proyectos tienen **un 60%** más de probabilidades de ser contratados en comparación con aquellos que solo mencionan sus habilidades sin verificación digital.



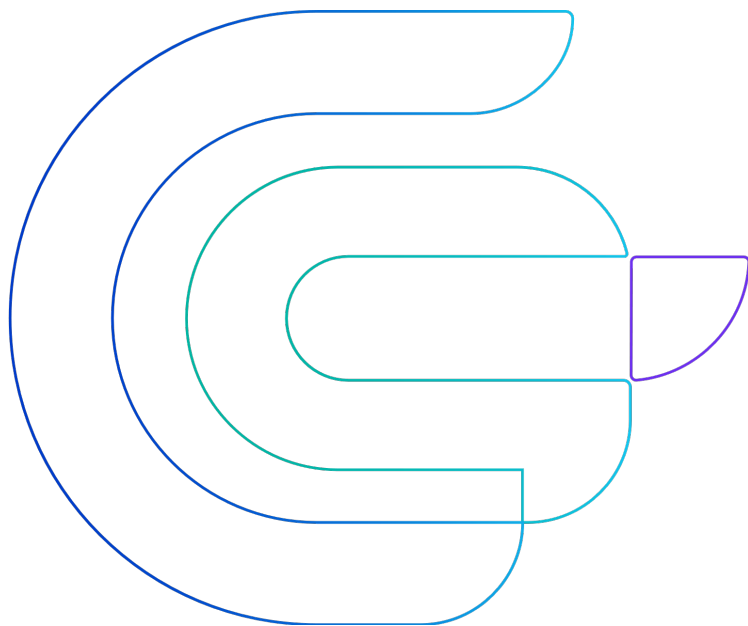
¿Por qué son importantes?



- **Empoderamiento de la Marca Personal:**

La visibilidad y verificación instantánea proporcionada por las insignias digitales permiten a los profesionales no solo demostrar sus habilidades, sino también construir una marca personal fuerte. Según un estudio de LinkedIn, los profesionales que utilizan insignias digitales tienen un 24% más de probabilidades de avanzar en sus carreras. La certificación y las insignias digitales no son solo una validación del conocimiento, sino también una herramienta poderosa para la mejora continua y la empleabilidad. En un mundo donde el aprendizaje permanente se ha convertido en la norma, estas credenciales son clave para el desarrollo profesional y la competitividad en el mercado laboral global.





No todas las insignias son iguales, y en **Certiprof**, estamos comprometidos con ofrecerte más que un simple reconocimiento digital. Al obtener una insignia emitida por certiprof, estarás recibiendo una validación de tu conocimiento respaldada por una de las entidades líderes en certificación profesional a nivel mundial.

Da el siguiente paso y obtén la insignia que te abrirá puertas y te posicionará como un experto en tu campo.



¿Por qué es importante obtener su certificado?

- **Prueba de experiencia:** Su certificado es un reconocimiento formal de las habilidades y conocimientos que ha adquirido. Sirve como prueba verificable de sus cualificaciones y demuestra su compromiso con la excelencia en su campo.
- **Credibilidad y reconocimiento:** En el competitivo mercado laboral actual, las empresas y los compañeros valoran las credenciales que le distinguen de los demás. Un certificado de una institución reconocida, como Certiprof, proporciona credibilidad instantánea e impulsa su reputación profesional.
- **Avance profesional:** Tener tu certificado puede abrirte las puertas a nuevas oportunidades. Ya se trate de un ascenso, un aumento de sueldo o un nuevo puesto de trabajo, las certificaciones son un factor diferenciador clave que los empleadores tienen en cuenta a la hora de evaluar a los candidatos.



¿Por qué es importante obtener su certificado?

- **Oportunidades de establecer contactos:** Poseer un certificado le conecta con una red de profesionales certificados. Muchas organizaciones cuentan con grupos de antiguos alumnos o de trabajo en red en los que puede compartir experiencias, intercambiar ideas y ampliar su círculo profesional.
- **Logro personal:** Obtener una certificación es un logro importante, y su certificado es un recordatorio tangible del trabajo duro, la dedicación y el progreso que ha realizado. Es algo de lo que puede sentirse orgulloso y mostrar a los demás.






Software Project Leader Professional Certification SPLPC

Issued by [Certiprof](#)

The holders of this badge are responsible for steering software development projects from inception to completion. They understand the importance of blending technical proficiency, leadership qualities, and project management skills. These candidates are not only adept in software development methodologies but also excel in leading teams, managing resources, and ensuring project milestones are met within time and budget constraints.

 Certification

 Paid

Skills

Adaptability

Analytical Thinking

Communication skills

Leadership Skills

Organizational Skills

<https://www.credly.com/org/certiprof/badge/software-project-leader-professional-certification-.1>



Aprendizaje Permanente

- Certiprof ha creado una insignia especial para reconocer a los aprendices constantes.
- Para el 2024, se han emitido más de 1,000,000 de estas insignias en más de 11 idiomas.

Propósito y Filosofía

- Esta insignia está destinada a personas que creen firmemente en que la educación puede cambiar vidas y transformar el mundo.
- La filosofía detrás de la insignia es promover el compromiso con el aprendizaje continuo a lo largo de la vida.

Acceso y Obtención de la Insignia

- La insignia de Lifelong Learning se entrega sin costo a aquellos que se identifican con este enfoque de aprendizaje.
- Cualquier persona que se considere un aprendiz constante puede reclamar su insignia visitando:

<https://certiprof.com/pages/certiprof-lifelong-learning>



Objetivos generales

1

Impulsar a los profesionales

Impulsar a los profesionales de la industria del software, fomentando un equilibrio integral de competencias en tres dimensiones: el **ecosistema** (fundamentos de software), **las habilidades blandas** y **las habilidades técnicas** para elevar su desempeño en el liderazgo.

2

Preparar para la certificación

Preparar a los participantes para que superen con éxito el examen de certificación, garantizando que estén equipados con el conocimiento y las habilidades necesarias para obtener el título de "Software Project Leader".



Objetivos de aprendizaje

1 Desarrollo integral de habilidades blandas para el liderazgo

Facilitar el crecimiento de los participantes en las habilidades blandas fundamentales (Comunicación, Organización y Motivación) para liderar proyectos de software de manera exitosa.

2 Desarrollo integral de habilidades técnicas para el liderazgo

Formar a los participantes en las habilidades duras fundamentales en el contexto de proyectos de software (efectividad, calidad y equipo).

3 Adaptación de profesionales de formación base no TI

Promover en los profesionales la comprensión y aplicación de conceptos técnicos fundamentales y de vanguardia para la industria del desarrollo de software, brindándoles una base sólida para su desempeño.

4 Crecimiento de profesionales de formación base TI

Impulsar a los profesionales en la transición y/o fortalecimiento de roles de liderazgo al acrecentar sus habilidades blandas y técnicas para el liderazgo, complementando sus competencias en software.

5 Conocimientos de vanguardia

Actualizar conocimientos y tendencias de la industria del software, en el uso de tecnologías, roles, métricas y habilidades clave para los retos vigentes.





¿Quién puede atender este entrenamiento?

Aquella persona que esté interesada en el liderazgo de proyectos de software, que requieren del entendimiento de conceptos centrales propios del ecosistema TI/Software, ampliar las habilidades de gestión y acrecentar las habilidades blandas, articulando estas tres dimensiones.



...

COMPARTE Y VERIFICA TUS LOGROS DE APRENDIZAJE FÁCILMENTE

#SPLPC #certiprof



 certiprof®

...

Introducción



¿Qué es el liderazgo?

Capacidad dinámica para dirigir la propia vida y/o equipos de personas logrando inspiración y motivación a través de una visión estratégica clara. Implica fomentar colaboración, sincronización y adaptación, con el propósito de alcanzar objetivos comunes y resultados óptimos.

"Warren Bennis y John C. Maxwell coinciden que el liderazgo es la capacidad de inspirar, guiar e influir a otros hacia la realización de una visión y objetivos compartidos."

Excelencia en el liderazgo



Implica

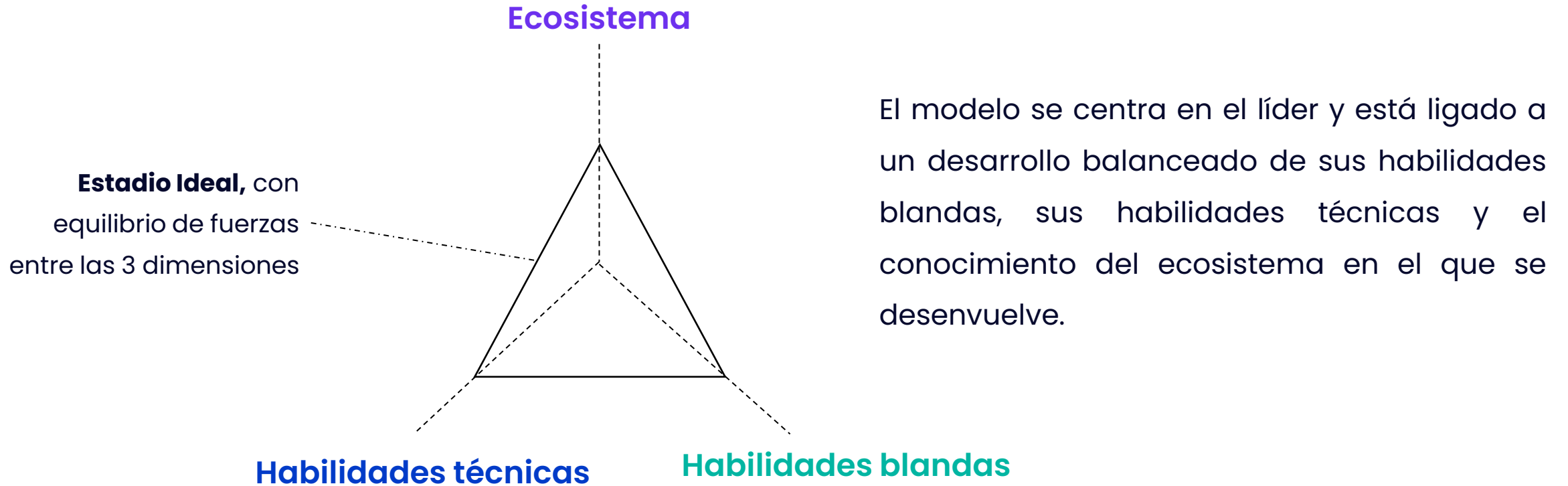
Capacidad dinámica para planificar, dirigir y tomar decisiones efectivas en contextos reales, cambiantes y desafiantes logrando resultados de valor, mientras se fomenta el crecimiento y el desarrollo propio, de los equipos y de la organización.

- Reconocimiento de recursos, capacidades y conocimientos propios.
- Lectura y dominio del contexto.
- Gestión de las interacciones para lograr resultado de valor.

Palacios, R. (2021) Transformación cultural en Colombia basada en valores. Estados Unidos de América



Modelo de excelencia en el desempeño del liderazgo



1. Lihtsus, 2023 basado en el Modelo de la Ocupación Humana (Moho). Kielhofner, 2004



Aplicación del *modelo* en líderes de proyectos de software



Ecosistema

- Fundamentos de software
- Desarrollo Web
- Desarrollo Móvil
- Cloud: Computación en la Nube

Habilidades blandas

- Comunicación
- Organización
- Motivación

Habilidades técnicas

- Efectividad
- Calidad
- Equipo

En este contexto se configura un **framework** que promueve el conocimiento y dominio de tres dimensiones, las cuales se movilizan positivamente entre sí para impulsar el liderazgo en proyectos de software: comprensión del **ecosistema** en términos de los conceptos fundamentales del software, tres **habilidades técnicas** claves y tres **habilidades blandas**. 2

Las dimensiones y sus elementos son una **base sólida** pero **no única** que impulsa la capacidad de desempeño del líder.

2. Lihtsus, 2023 basado en el Modelo de la Ocupación Humana (Moho). Kielhofner, 2004



Habilidades blandas



Valor de las habilidades blandas

Son el soporte humano para construir relaciones sólidas, generar influencia sostenible y promover el compromiso contribuyendo en la generación de una cultura organizacional propositiva y visionaria.



"Las softskills son lo que diferencia a los líderes sobresalientes. La capacidad de inspirar, comunicar y colaborar es esencial para construir equipos fuertes y llevar a cabo proyectos ambiciosos."

Sheryl Sandberg

COO Facebook y fundadora de Leanin.org

"En el mundo empresarial, las habilidades sociales son igualmente importantes que las habilidades técnicas. Las personas con fuertes habilidades de comunicación y empatía tienden a liderar equipos más eficaces y crear un ambiente de trabajo positivo."

Richard Branson

Fundador de Virgin Group

- 360 empresas



Habilidades blandas para el liderazgo en proyectos de software

Según Goleman, D. (1998), el éxito de los proyectos depende en un 75% de las habilidades blandas pues son la base humana y relacional que requieren los equipos para capitalizar aprendizaje, sentido de colaboración, adaptación y efectividad sobre los resultados deseados.²

López, S. (2020) considera que existen tres tipos de habilidades en el liderazgo de proyectos de software.³

2. Siu, J. L. R. (2020). Las habilidades blandas como base del buen desempeño del docente universitario. Innova research journal, 5(2), 186-199.

Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=7475508>

3. López, S. (2020). Análisis de los requerimientos en habilidades de la industria del software para gerentes de proyectos en entornos ágiles.

Desarrollo e Innovación en Ingeniería, 192. Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=8708578>



Habilidades blandas para el liderazgo en proyectos de software

1

Deseables

Son bien valoradas y pueden marcar la diferencia en el desempeño del líder.

1. Toma de decisiones
2. Pensamiento Crítico
3. Multicultural
4. Confianza
5. Adaptación

2

Requeridas

Son necesarias para desempeñar con éxito las responsabilidades asignadas y cumplir con los estándares de rendimiento.

1. Colaboración
2. Relaciones interpersonales
3. Resolución de problemas
4. Resolución de conflictos
5. Orientación

3

Mandatorias

Son esenciales y deben estar presentes en el líder.

1. **Comunicación**
2. **Organización**
3. Motivación

Basic soft skills

SKCLS

Gnarr eciout
lohmven youre
onr duaty

+

+

Intermediate

SKLLS

Guce fun d'ea diaom
fororthenx
cogf t skills

+

+

Advanced soft skills

SKLLS

Errarent t euialomed
ono fushe anrcting
mevenrcingans



...

Comunicación



SPLPC™ Versión 032025

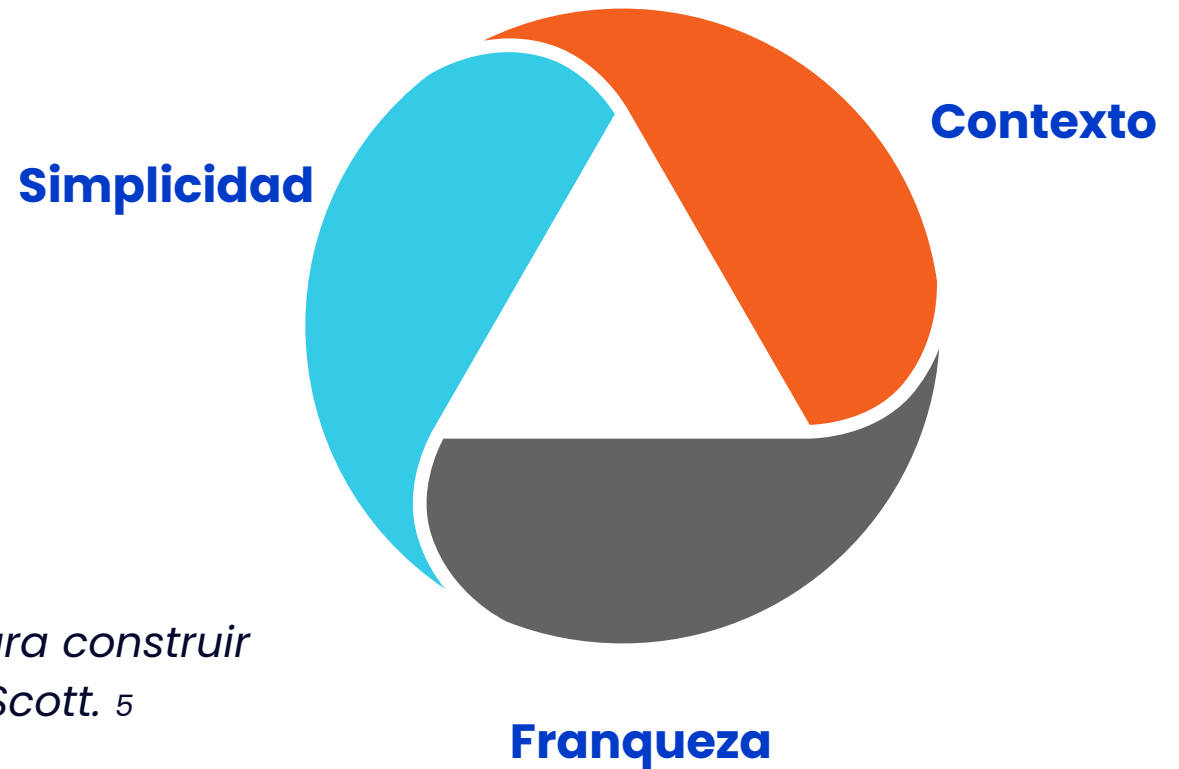


Comunicación

Es un proceso complejo de interacción donde se comparten ideas, visiones, sentimientos con propósitos específicos.

Dentro de los aspectos centrales para una comunicación de calidad se destaca el fomentar interacciones **simples, francas y contextuales**.⁴

*La comunicación auténtica y sincera es esencial para construir relaciones sólidas en las organizaciones. Kim Scott.*⁵



4. Yang, Y., Saladrigas Medina, H., & Torres Ponjuán, D. (2016). El proceso de la comunicación en la gestión del conocimiento. Un análisis teórico de su comportamiento a partir de dos modelos típicos. Revista universidad y sociedad, 8(2), 165-173. Recuperado de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202016000200022

5. Forbes (2021) El Candor Radical: de habilidades blandas a esenciales
Recuperado de <https://www.forbes.com.mx/red-forbes-el-candor-radical-de-habilidades-blandas-a-esenciales/>



Aspectos centrales ⁶



Simplicidad

Es la habilidad para compartir información de manera clara y precisa asegurando su fácil comprensión.



Franqueza

Es una práctica que busca el intercambio de información con base en la transparencia y la sinceridad.



El contexto en la comunicación

Se relaciona con el conjunto de circunstancias y condiciones donde suceden las interacciones comunicativas.

"Si es posible expresar un pensamiento de forma más sencilla, es necesario hacerlo. La sencillez es amiga de la buena comunicación." George Orwell

6. Caridad, M., Batista, J., & Castellano, M. I. (2007). Comunicación revisitada en el contexto de las organizaciones empresariales. *Clío América*, 1(2), 240-258. Recuperado de <https://dialnet.unirioja.es/descarga/articulo/5114837.pdf>





Aspectos centrales

1 Simplicidad

Lograr **simplicidad** en la comunicación impulsa un liderazgo preciso y eficaz.

2 Franqueza

Apropiar interacciones basadas en la **franqueza** moviliza liderazgos con credibilidad y que generan confianza.

3 Contexto

Impulsar una comunicación en **contexto** eleva un liderazgo asertivo y exitoso.

Estos aspectos se configuran como un marco para el desarrollo de habilidades comunicativas en los líderes.

"La comunicación siempre tiene en cuenta el contexto y las necesidades de la audiencia." Stephen R. Covey



...

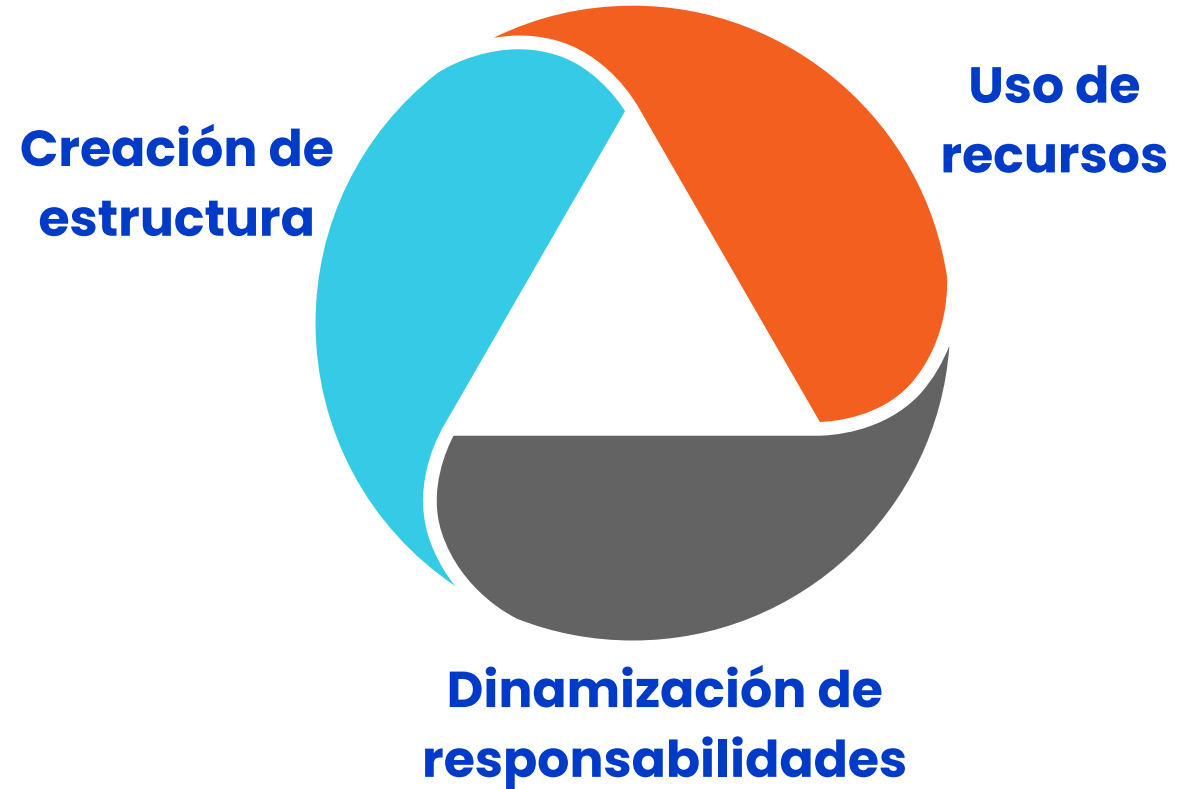
Organización



Organización

Es la habilidad para establecer patrones estructurados de pensamiento que permiten guiar procesos y acciones de acuerdo con el rol que se asume.⁷

La base para el desarrollo de la organización en los líderes se fundamenta en la creación de **estructura**, el uso óptimo de **recursos**, y la dinamización de **responsabilidades**.



7. Kielhofner, G. (2004). *Terapia ocupacional. Modelo de Ocupación Humana. Teoría y aplicación*. Ed. Médica Panamericana. Recuperado de <https://books.google.com.pe/books?id=hBQfhHCs-XIC&printsec=frontcover#v=onepage&q&f=false>





Aspectos centrales 8



Creación de estructura

Se relaciona con la habilidad para ordenar procesos, actividades y recursos de forma lógica y congruente.



Dinamización de responsabilidades

Es la habilidad para identificar, asignar y hacer seguimiento a las actividades y compromisos propios y de otros.



Uso de recursos

Es la habilidad para identificar y usar asertivamente los recursos técnicos y materiales necesarios en un proyecto.

8. Covey, S. R. (2015). Los 7 Hábitos de la Gente Altamente Efectiva: Edición de Imágenes. Mango Media Inc. Recuperado de <http://200.70.33.130/images2/SGHPE/7%20HABITOS.pdf>





Aspectos centrales

1

Estructura

Una **estructura** de pensamiento organizada con acciones definidas y coherentes favorecen un liderazgo enfocado y con dirección.

2

Responsabilidades

Una óptima gestión de las **responsabilidades** promueve un liderazgo organizado, reconocedor de capacidades y orientado hacia la acción con sentido.

3

Recursos

Un liderazgo que gestiona y hace uso de los **recursos** asegura sólidos soportes para impulsar el cumplimiento de objetivos y metas.

Estos aspectos se configuran como un marco para incrementar habilidades centrales de organización en un líder



Motivación



Motivación

Es la habilidad que impulsa la realización de actividades de valor con **compromiso**, máximo **desempeño** y **disfrute** por lo que se realiza (Kielhofner, 2004)⁹

"El liderazgo comienza con la voluntad de motivar a otros a dar lo mejor de sí mismos." Bill Bradley



9. Kielhofner, G. (2004). *Terapia ocupacional. Modelo de Ocupación Humana. Teoría y aplicación*. Ed. Médica Panamericana. Recuperado de <https://books.google.com.pe/books?id=hBQfhHCs-XIC&printsec=frontcover#v=onepage&q&f=false>



Aspectos centrales 10



Desempeño

Es el resultado de la interacción entre la persona, un contexto y una actividad.



Disfrute y sentido de logro

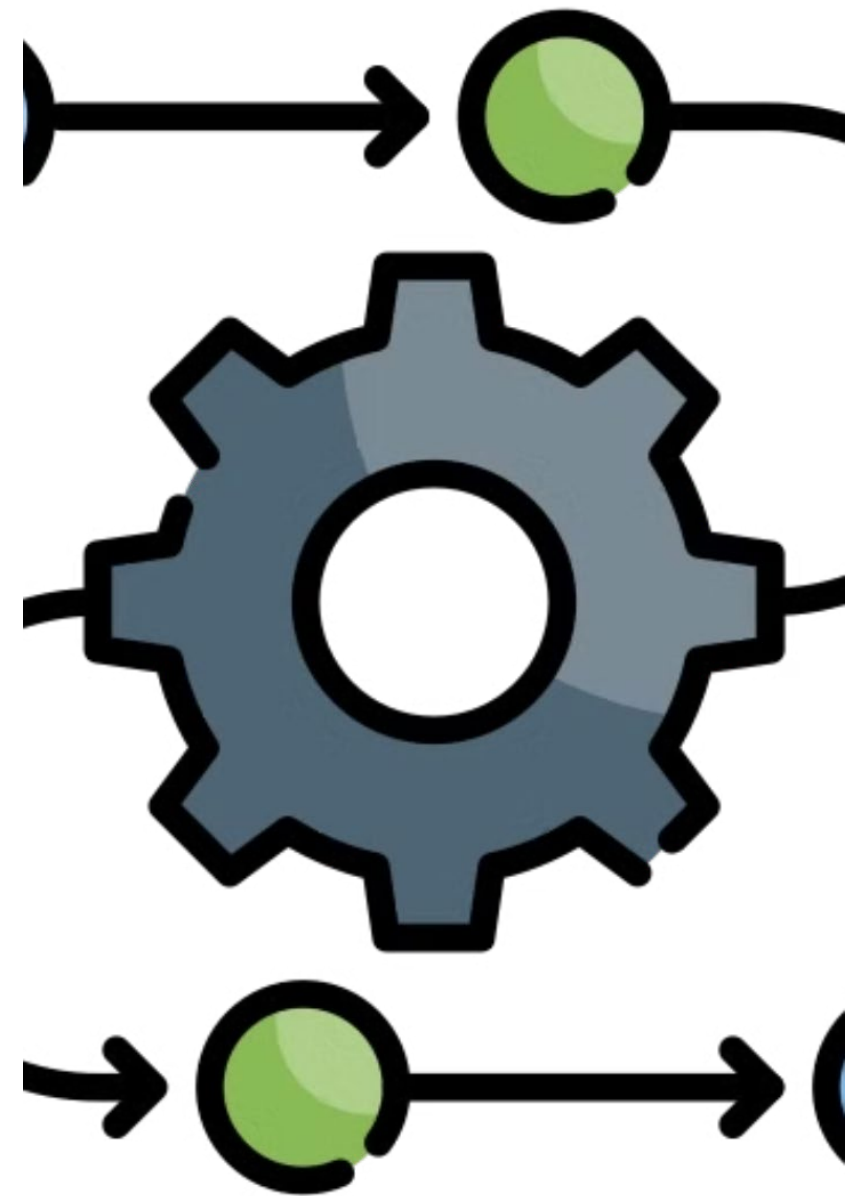
Es la sensación de placer y satisfacción que se experimenta cuando se realizan actividades de valor y cuando se alcanza con éxito los propósitos.



Compromiso

Es la habilidad para lograr conexión y dedicación con los propósitos y actividades en tiempos y límites establecidos.

10. Solano, A. C., & Minervino, R. (2007). Motivación para liderar, inteligencia práctica y efectividad de los líderes. Revista Iberoamericana de Diagnóstico y Evaluación-e Avaliação Psicológica, 1(23), 175-199. Recuperado de <https://www.redalyc.org/pdf/4596/459645446010.pdf>





Aspectos centrales

1

Desempeño

Un **desempeño** constante y satisfactorio, favorece la confianza y la capacidad del líder para asumir actividades diarias y atender desafíos permanentes.

2

Disfrute

Lograr el **disfrute** por lo que se hace, moviliza liderazgos que apuestan por la felicidad y el bienestar mientras se alcanzan más y mejores resultados.

3

Compromiso

Un **compromiso** decidido y eficaz genera un liderazgo conectado con los procesos y las visiones.

Estos aspectos se configuran como un marco para aumentar la motivación como habilidad progresiva en el liderazgo.



Fundamentos de Software



Desarrollo Web



Desarrollo Web

Es una disciplina que se enfoca en el proceso de construir y mantener sitios web. Esta práctica abarca una amplia variedad de tareas, que van desde el **diseño** y la **codificación** hasta el **mantenimiento** y la **actualización**.¹¹

11. GeeksforGeeks. (2023). Web Development - GeeksforGeeks. [online]. Recuperado de <https://www.geeksforgeeks.org/web-development/>



Importancia

El **desarrollo web** es una disciplina esencial en el mundo actual. Los sitios web son una parte integral de la vida cotidiana, y se utilizan para una amplia gama de propósitos, en los cuales se encuentran: la educación, el entretenimiento, el comercio electrónico, los servicios gubernamentales, entre otros.

Desarrollo Web ¹²

Áreas

El desarrollo web es un campo amplio que se puede dividir en diferentes áreas, cada una con sus propias especialidades y enfoques, dentro de las que se destacan:

- ✓ Diseño Web
- ✓ Desarrollo Front-end
- ✓ Desarrollo Back-end
- ✓ Desarrollo de Bases de datos
- ✓ Desarrollo de APIs
- ✓ Arquitectura de Software

12. Tekla (2022). ¿Qué es el desarrollo web? Recuperado de <https://tekla.io/blog/que-es-desarrollo-web/>



Front-end





Front-end

También conocido como **Client Site** (lado del cliente), es la **parte visible** de una aplicación o de un sitio web que se encarga de la interactividad con los usuarios.

Foto de [Austin Distel](#) en Unsplash



Tecnologías

Existen tres tecnologías fundamentales que trabajan juntas para crear la experiencia que un usuario tiene en la web, desde su estructura básica hasta su apariencia y comportamiento interactivo.



HTML

(HyperText Markup Language)

El **lenguaje de marcado de hipertexto** es el esqueleto de una página web. Se utiliza para estructurar y organizar el contenido de una página como los títulos, los párrafos, las imágenes, los enlaces, entre otros.



CSS

(Cascading Style Sheets)

Las **hojas de estilo en cascada** se utilizan para darle estilo y diseño a una página web. CSS permite cambiar colores, fuentes, tamaños y disposición de elementos en la página para que se vea atractiva y coherente.¹³



JavaScript

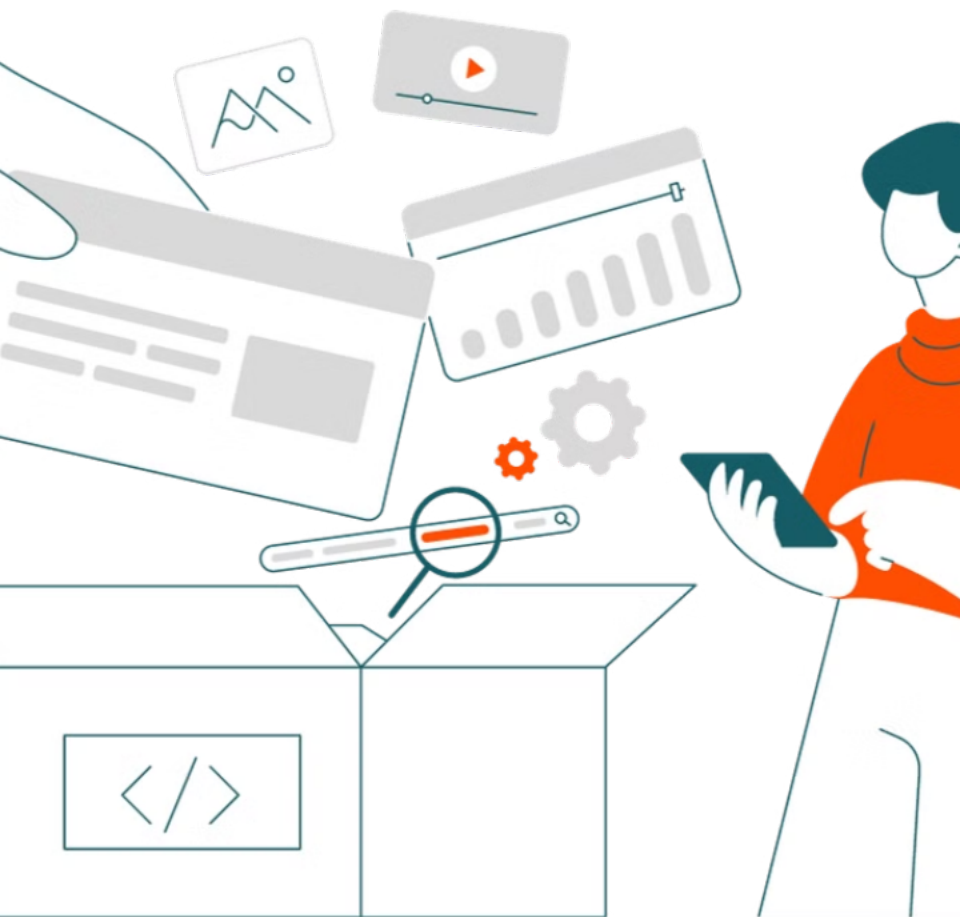
Es un **lenguaje de programación** que permite la interactividad en una página web. Con JavaScript, se pueden realizar tareas como validar formularios, crear animaciones y, en general, hacer que la página responda y sea dinámica.¹⁴

13. Digital IONOS (s.f) ¿Qué es CSS? Definición y aplicación. Recuperado de <https://www.ionos.es/digitalguide/paginas-web/disenio-web/que-es-css/>

14. Medium, Multimedia. (2023). Guía para encontrar programas de estudio en Desarrollo Web en España. Recuoerado de <https://www.mediummultimedia.com/web/donde-estudiar-desarrollo-web-en-espana/>



Framework Web



1

Definición

Es un conjunto de herramientas, reglas y bibliotecas que permiten a los desarrolladores, acelerar la construcción de una aplicación web al proporcionar componentes y funcionalidades comunes. 15

2

Componentes

Los frameworks suelen incluir código **HTML**, **CSS** y **JavaScript** predefinido, lo que reduce la complejidad del desarrollo y permite a los desarrolladores centrarse en la lógica del sitio en lugar de tener que crear estos componentes desde cero. Esto permite el desarrollo de sitios web de manera eficiente y estructurada.

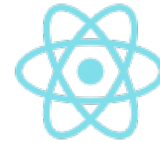
15. Wam Glowth Agents (2023). Frameworks en el desarrollo web: las mejores prácticas para tu negocio online Recuperado de <https://www.wearemarketing.com/es/blog/frameworks-en-el-desarrollo-web-las-mejores-practicas-para-tu-negocio-online.html>



Frameworks front-end

Existe una amplia gama de frameworks front-end que permiten a los desarrolladores elegir la herramienta que mejor se adapte a sus necesidades y proyectos. Entre los más populares se encuentran **React**, **Vue.js** y **Angular** ¹⁶

Estos frameworks están basados en **JavaScript**, y ofrecen un sólido soporte para TypeScript, un lenguaje de programación tipado que brinda beneficios adicionales en términos de mantenibilidad y detección de errores en el código. ¹⁷



React



Vue.js



Angular

Cada uno de estos frameworks tiene enfoques y características diferentes. La elección del framework adecuado dependerá de las necesidades específicas del proyecto, los objetivos comerciales, las preferencias de desarrollo, etc.

16. StackOverflow. (2023). Encuesta sobre tecnologías y frameworks web 2023. Recuperado de <https://survey.stackoverflow.co/2023/#technology>

17. GeeksforGeeks. (2023). Top Front-End Frameworks in 2023. Recuperado de <https://www.geeksforgeeks.org/top-front-end-frameworks/>



Back-end





Imagen de senivpetro en Freepik

Back-end

También conocido como **Server Site** (lado del servidor), es la **parte no visible** de una aplicación web que se encarga del **procesamiento de datos**, su **almacenamiento** y la gestión de la **lógica de negocio**.

El back-end se comunica con el front-end para proporcionar datos y funcionalidades. Es responsable de garantizar que las solicitudes del cliente se procesen de manera adecuada y de manejar la seguridad y la gestión de la base de datos.



Tecnologías

Existe un gran número de lenguajes de programación de propósito general. Entre ellos destacan:

- **Java:** Aplicaciones móviles, desarrollo de software multiplataforma.
- **Python:** Análisis de datos, inteligencia artificial y aprendizaje automático.
- **PHP:** Sitios web dinámicos, sistemas de gestión de contenidos (CMS).
- **Ruby:** Aplicaciones web de bases de datos.
- **JavaScript con Node.js:** Aplicaciones web de alto rendimiento, como chats en línea, aplicaciones de transmisión en vivo, servidores de juegos.¹⁹

Se han enunciado aquí enfoques comunes que en ocasiones tienen estos lenguajes.

19. StackOverflow. (2023). Encuesta sobre tecnologías y frameworks web 2023. Recuperado de <https://survey.stackoverflow.co/2023/#technology>

Lenguajes

Frameworks



Java



Spring



Python



django

Django



php



Laravel



Ruby



Ruby on Rails



JavaScript



express

Express.js



Tecnologías

Asociado a cada lenguaje de programación, existe uno o varios frameworks que facilitan el desarrollo al ofrecer soluciones comunes a desafíos recurrentes. ²⁰

De la misma manera que el front-end, la elección del framework depende de las necesidades específicas de cada proyecto. ²¹

A continuación, se listan lenguajes de programación destacados con el framework asociado más popular.

20. StackOverflow. (2023). Encuesta sobre tecnologías y frameworks web. Recuperado de <https://survey.stackoverflow.co/2023/#technology>

21. GeeksforGeeks. (2023). 10 Best Web Development Frameworks to Use in 2023. Recuperado de <https://hackr.io/blog/web-development-frameworks>

Lenguajes

Frameworks



Java



Spring



Python

django

Django



php



Laravel



Ruby



Ruby on Rails



JavaScript

express

Express.js



Bases de Datos



“Una base de datos es una colección organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático.” ²³

Permiten la recuperación eficiente de información, la realización de consultas, la actualización de registros y el análisis de datos, lo que facilita la toma de decisiones informadas.

Existen dos tipos principales de bases de datos: **SQL (relacionales)** y **NoSQL (no relacionales)**.

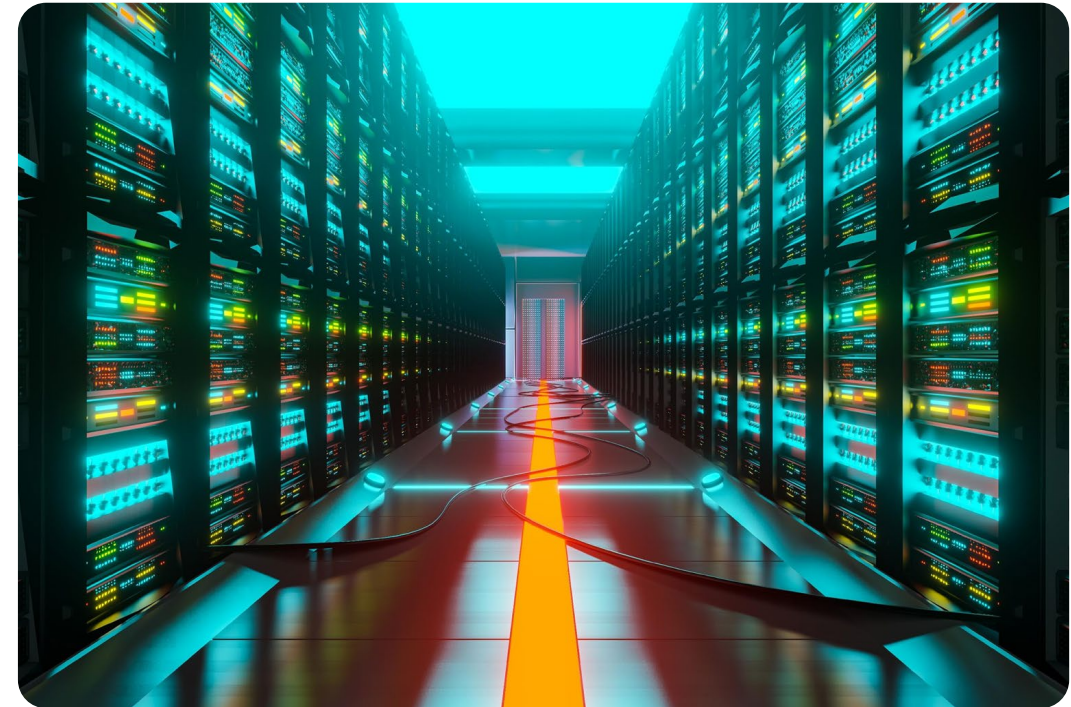


Imagen de DCStudio en Freepik

23. Oracle. (s.f.). ¿Qué es una base de datos? Oracle. Recuperado de <https://www.oracle.com/database/what-is-database.html>

Tipos de Bases de Datos

Relacionales SQL

Almacenan datos de manera organizada, similar a las tablas en una hoja de cálculo. Cada fila representa un registro y cada columna contiene un tipo de información.

User Info				Address Info			
KEY	First	Last	ZIP_id	ZIP_id	CITY	STATE	ZIP
1	Frank	Weigel	2	1	DEN	CO	30303
2	Ali	Dodson	1	2	MV	CA	94040
3	Mark	Axed	3	3	CHI	IL	60609
4	Steve	Yen	4	4	NY	NY	10010

+

Si se requiere:

- ✓ Estructura de datos estable
- ✓ Transacciones y consistencia
- ✓ Consultas complejas y análisis
- ✓ Integridad de los datos ²⁴.

← **¿Cuándo utilizarlo?** →

No Relacionales No Sql

Almacenan y gestionan datos no estructurados o semiestructurados, como documentos, fotos y videos, datos de redes sociales, entre otros.

```
JSON
1 {
2   "ID": 1,
3   "FIRST": "Frank",
4   "LAST": "Weigel",
5   "ZIP": "94040",
6   "CITY": "MV",
7   "STATE": "CA"
8 }
9
```

Si se requiere:

- ✓ Escalabilidad y rendimiento
- ✓ Flexibilidad del esquema
- ✓ Réplica de los datos
- ✓ Rentabilidad ²⁵

24. Amazon Web Services. (s.f.). Aspectos importantes de las bases de datos relacionales[en línea]. Recuperado de <https://aws.amazon.com/es/relational-database/>

25. IBM. (s.f.). Ventajas de NoSQL [en línea]. Recuperado de <https://www.ibm.com/mx-es/topics/nosql-databases>



Relacionales

Utilizan el lenguaje **estándar** SQL (Structured Query Language) para gestionar datos. Permite realizar diversas operaciones en una base de datos, como la inserción, actualización, eliminación y recuperación de datos. ²⁶



26. Amazon Web Services. (s.f.). ¿Qué son los estándares de SQL?. Recuperado de <https://aws.amazon.com/es/what-is/sql/>

27. IBM. (s.f.). Tipos de bases de datos NoSQL. Recuperado de <https://www.ibm.com/es-es/topics/nosql-databases>

No Relacionales

No tienen un lenguaje universal. En su lugar, utilizan APIs y consultas específicas para cada tipo de base de datos no relacional. Estas bases de datos son más flexibles en cómo almacenan y acceden a los datos. ²⁷



Tipos de Bases de Datos No Relacionales :

- ✓ **Documentales:** Almacenan datos en documentos, como hojas de cálculo o archivos JSON.
- ✓ **Columnares:** Almacenan datos en columnas en lugar de filas, optimizadas para análisis de datos.
- ✓ **Clave-Valor:** Almacenan datos en pares clave-valor, donde un valor está asociado a una clave única.
- ✓ **Grafos:** Representan datos como puntos interconectados por relaciones.
- ✓ **Búsqueda:** Almacenan y organizan datos para búsquedas eficientes y análisis de texto.



Tecnologías

Sistemas de Gestión de Bases de Datos Relacionales 28

Un Sistema de Gestión de Bases de Datos Relacionales (RDBMS) es un software diseñado para administrar y gestionar bases de datos relacionales. Estas bases de datos almacenan datos en tablas con filas y columnas.

Los RDBMS proporcionan un conjunto de herramientas y funcionalidades para crear, mantener y consultar estos datos de manera eficiente. Entre los más populares se encuentran:

28. StackOverflow. (2023). Encuesta sobre tecnologías y frameworks web 2023. Recuperado de <https://survey.stackoverflow.co/2023/#technology>



Soporta tipos de datos complejos y funciones avanzadas



Destaca por velocidad y escalabilidad



Ligero y sin servidor, ideal para almacenamiento local



Integración con productos y herramientas de Microsoft



Ampliamente utilizado en aplicaciones empresariales, conocido por su rendimiento y escalabilidad



Tecnologías

Sistemas de Gestión de Bases de Datos No Relacionales

29

Un Sistema de Gestión de Bases de Datos No Relacionales (NoSQL) es un software que administra bases de datos con una estructura de datos más flexible que las bases de datos relacionales tradicionales. Esto les permite adaptarse a una amplia gama de aplicaciones y necesidades.

Cada sistema NoSQL tiene su propia forma de comunicación, lo que brinda flexibilidad en la elección del sistema que mejor se adapte a un proyecto específico. Entre los más populares se encuentran:

29. StackOverflow. (2023). Encuesta sobre tecnologías y frameworks web 2023. Recuperado de <https://survey.stackoverflow.co/2023/#technology>



Sistema enfocado en el modelo documental



Sistemas en tiempo real enfocados en el modelo documental



Sistemas enfocados en el modelo clave-valor



Sistema enfocado en el modelo de búsqueda



Sistema enfocado en el modelo de grafos





API



API

Application Programming Interface

Una **Interfaz de programación de aplicaciones**, es un conjunto de reglas y protocolos que permiten a dos componentes de software comunicarse y compartir datos entre sí. Proporciona una interfaz consistente y estructurada para interactuar con un sistema o servicio. ³⁰

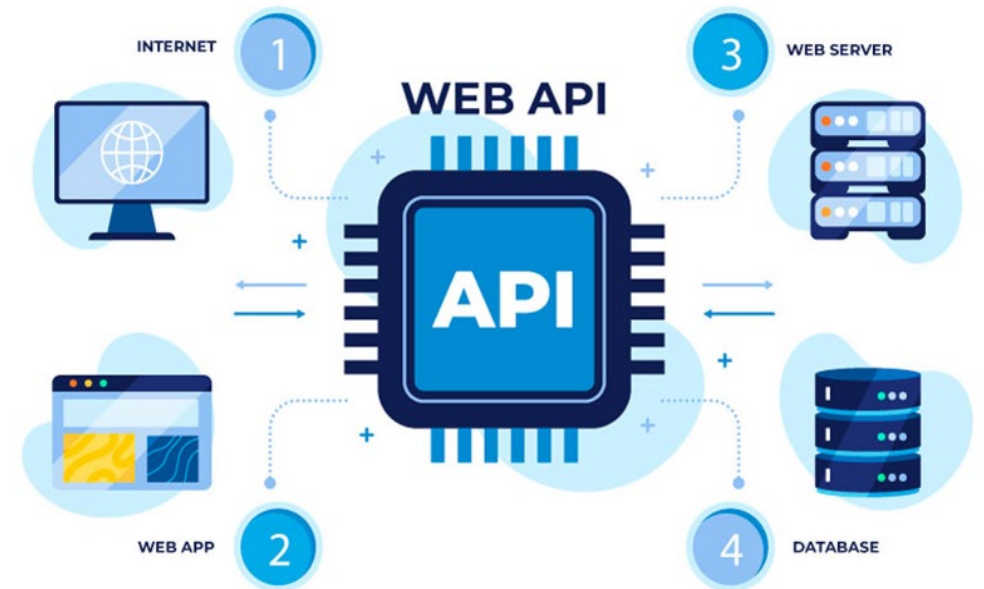


Imagen de Freepik

30. AWS. (s.f.) ¿Qué es un API? AWS. Recuperado de <https://www.oracle.com/database/what-is-database.html>



Tipos de APIs

Locales

Las APIs locales son aquellas que se utilizan para comunicarse con componentes o recursos en el mismo sistema o dispositivo donde se encuentra la aplicación que las invoca. Estas APIs no implican la comunicación a través de una red o internet. Algunos ejemplos de APIs locales incluyen:

- ✓ **APIs de sistema operativo:** Para acceder a funciones del sistema, como el sistema de archivos o la cámara de un dispositivo móvil.



- ✓ **APIs de base de datos local:** Para interactuar con bases de datos locales, como SQLite en aplicaciones móviles.



31. W3C. (s.f.). Servicio web, [online]. Recuperado de <https://www.w3.org/2002/ws/>

32. TechTarget. (s.f.). SOAP [online]. Recuperado de <https://searchmicroservices.techtarget.com/definition/SOAP>

33. Abi.gitbook.io. (s.f.). Servicios REST. [online]. Recuperado de <https://abi.gitbook.io/net-core/3.-servicios-rest/3.1-servicios-rest>



Tipos de APIs

Remotas

Las APIs remotas permiten la comunicación entre aplicaciones o sistemas a través de un:

Servicio Web

Componente de software que se expone a través de Internet o una red local para permitir el intercambio de datos entre diferentes sistemas. ³¹

Dos enfoques comunes para la implementación de servicios web son:

SOAP

Simple Object Access Protocol

Protocolo de comunicación utilizado para intercambiar información estructurada en aplicaciones web. ³²

El formato de mensaje utilizado es XML



REST

Representational State Transfer

Estilo arquitectónico que utiliza el protocolo HTTP para la comunicación entre aplicaciones web. ³³

El formato de mensaje utilizado es JSON



Tecnologías

Para crear APIs

Se emplean los lenguajes de programación Back-end anteriormente mencionados, que incluyen Node.js, Java, Ruby, Python, entre otros.

Para probar APIs

Entre los más populares están **Postman** y **SoapUI** para pruebas de APIs tanto REST como SOAP.



Para documentar APIs

Una de las tecnologías más populares es **Swagger**, que permite definir la estructura y documentación de una API, a través de herramientas como **Swagger UI** para mostrar la documentación.



...

Sistemas de Control de Versiones

SPLPC™ Versión 032025



Sistema de control de versiones

Version Control Systems

Los sistemas de control de versiones son herramientas que **registran cambios en archivos** y permiten a los desarrolladores **colaborar** de manera eficiente en proyectos de software.

Facilitan la gestión de múltiples versiones del código y generar trazabilidad de cada cambio a lo largo del **tiempo**. Ayudan a evitar conflictos y permiten volver a versiones anteriores si es necesario. ³⁴

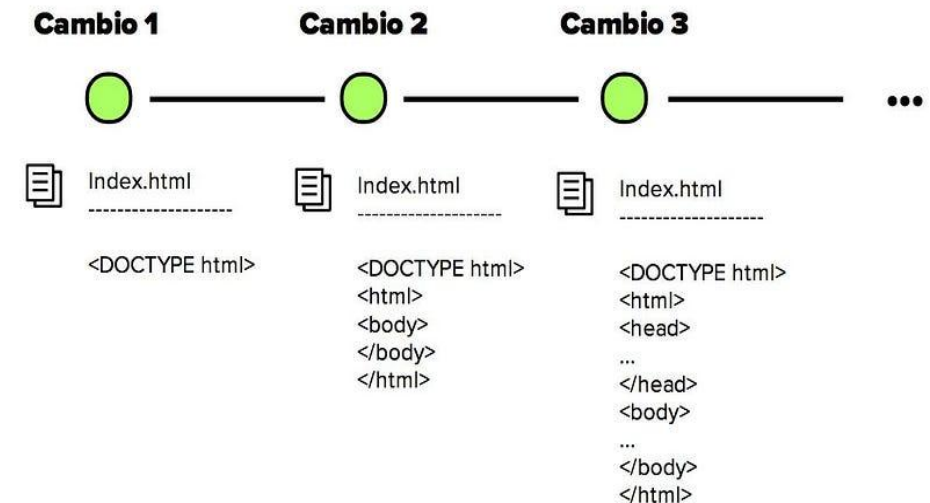


Imagen de @jointdeveloper en [Medium](#)

34. Atlassian. (s.f) ¿Qué es el Control de Versiones? Recuperado de <https://www.atlassian.com/es/git/tutorials/what-is-version-control>



Tipos de Sistemas de control de versiones ³⁵

Centralizados

Es un sistema en el que una única copia del repositorio de código fuente está almacenada en un servidor central.

Los desarrolladores trabajan en sus copias locales y deben comunicarse con el servidor central para obtener o enviar cambios. Los ejemplos incluyen CVS y Subversion (SVN).

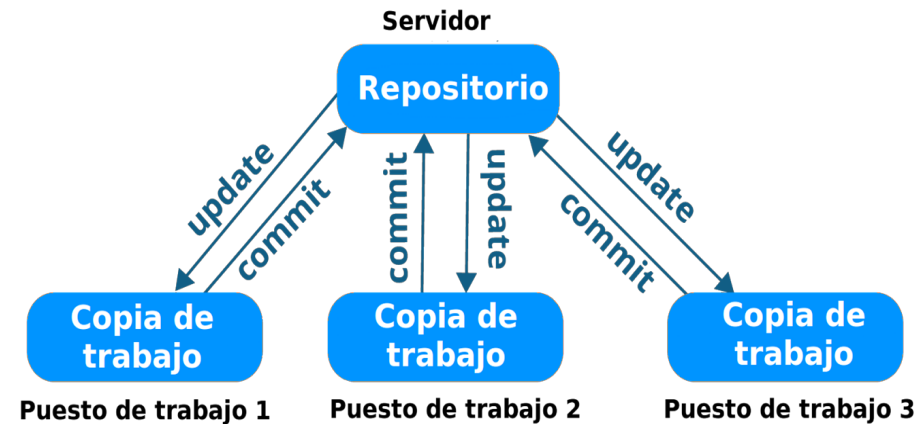


Imagen de Opentix

Tecnologías



35. Atlassian Bitbucket. (s.f). Software de control de versiones para equipos profesionales. Recuperado de <https://bitbucket.org/product/es/version-control-software>



Tipos de Sistemas de control de versiones ³⁵

Distribuidos

Es un sistema en el que cada desarrollador tiene una copia completa del repositorio, incluyendo todo el historial de versiones.

Los desarrolladores pueden trabajar de manera independiente en sus copias locales y compartir cambios directamente entre ellas. Los ejemplos incluyen Mercurial y Git.

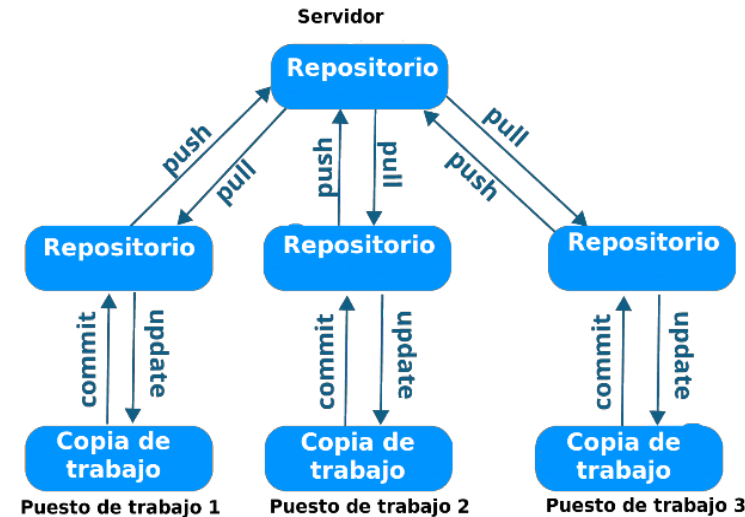


Imagen de Opentix

Tecnologías



35. Atlassian Bitbucket. (s.f). Software de control de versiones para equipos profesionales. Recuperado de <https://bitbucket.org/product/es/version-control-software>



Git

Git es resaltado como el sistema de control de versiones **estándar** en la actualidad debido a su capacidad para gestionar múltiples **repositorios remotos**, permitiendo un control preciso de las versiones del código y facilitando la colaboración eficiente en el entorno de desarrollo. 36

Plataformas de alojamiento de repositorios remotos

Proporcionan un entorno para alojar, gestionar y colaborar en proyectos de desarrollo de software utilizando Git como sistema de control de versiones. Entre los más populares se encuentran:



La elección entre ellas depende de las necesidades específicas del equipo o la organización.

36. Acens for Telefónica Company (2015). Whitepaper: Control de Versiones GIT y GITHUB. Recuperado de <https://www.acens.com/comunicacion/white-papers/control-versiones-git-github/>



...

Arquitectura de Software



SPLPC™ Versión 032025



Arquitectura de Software

Es el conjunto de **estructuras** necesarias para **razonar** sobre el sistema, que comprenden **elementos** de software, **relaciones** entre ellos y **propiedades** de ambos.

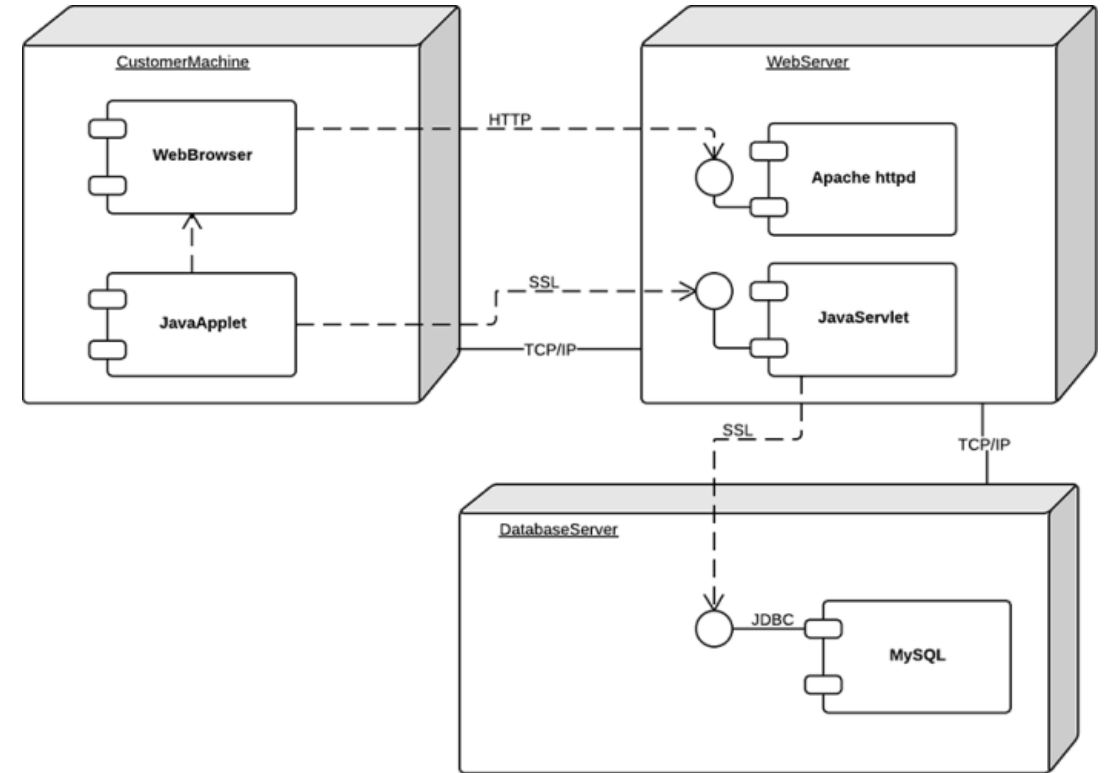


Diagrama de Implementación de Lucidchart.2.

37. Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice. Addison-Wesley.

38. Lucidchart. (s.f.). Diagrama de Implementación. Recuperado de <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>

Arquitectura de Software

Los siguientes son elementos esenciales que contribuyen a la definición y construcción de una arquitectura de software de un sistema:

1 Diseño de alto nivel

Es el proceso de establecer una visión general y una estructura principal para el sistema de software. Implica la definición de los componentes clave del sistema, cómo interactúan entre sí y cómo se organizan para cumplir con los requisitos del proyecto.

2 Estructura (patrones/tipos de arquitectura)

La elección de la estructura arquitectónica es uno de los primeros y más críticos pasos en el diseño de un sistema de software. Implica seleccionar un patrón de arquitectura apropiado, como el patrón de capas, microservicios, arquitectura hexagonal, entre otros.

3 Atributos de calidad

Son las características críticas que definen el éxito del sistema en términos de rendimiento, seguridad, escalabilidad, entre otros.

4 Documentación

Consiste en crear una guía clara y suficiente que describe la estructura, los componentes, las relaciones y los atributos de calidad del sistema. Esta documentación actúa como un mapa que orienta la construcción del sistema en todo el ciclo de desarrollo y por lo general es realizada utilizando el Lenguaje Unificado de Modelado (UML).

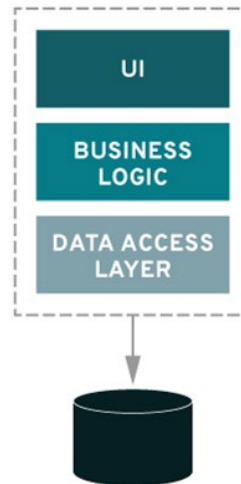


Patrones de arquitectura

Los patrones de arquitectura son soluciones generales a problemas comunes en la arquitectura de software. Estos patrones proporcionan un enfoque estructurado y reutilizable para resolver situaciones recurrentes en el desarrollo de software. ³⁹ Algunos patrones comunes son:

Arquitectura Monolítica

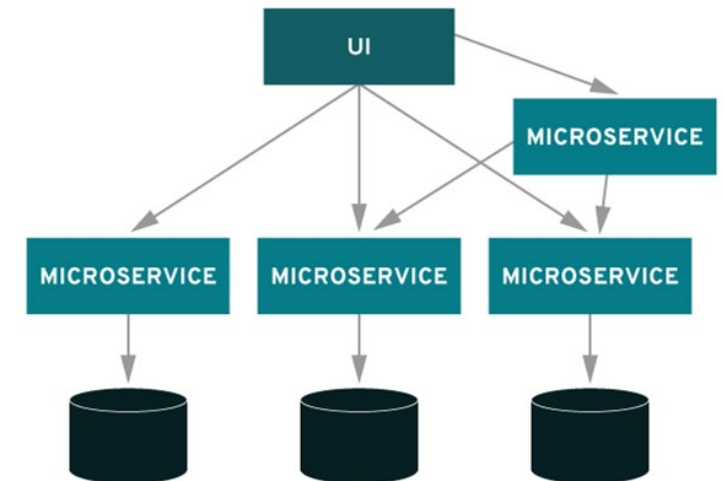
Una única aplicación donde todos los componentes (interfaz de usuario, la lógica de negocio y la base de datos) están integrados en un solo código base.



Arquitectura Monolítica .39.

Arquitectura de Microservicios ⁴⁰

Divide una aplicación en un conjunto de servicios independientes y pequeños que se comunican entre sí a través de API



Arquitectura de Microservicios.39.

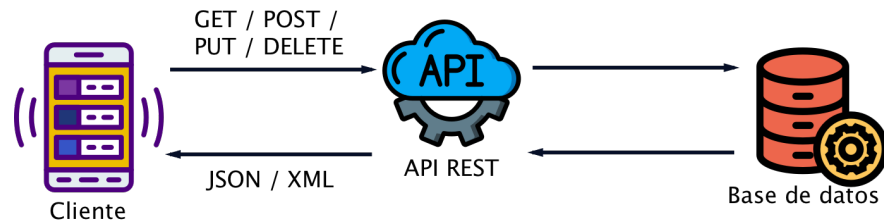
39. Medium. (s.f.). ¿Microservicios?. Recuperado de <https://medium.com/@llevit/microservicios-e8c71a6efe52>



Patrones de arquitectura

Arquitectura Cliente-Servidor

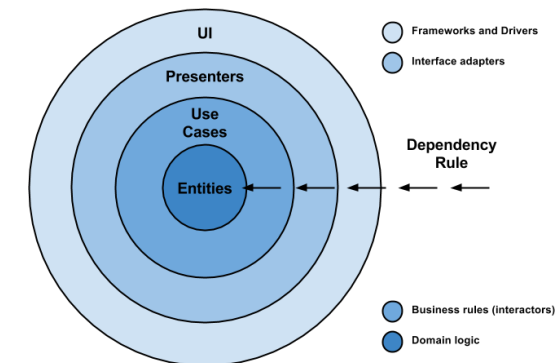
Divide la aplicación en dos partes: el cliente (que solicita servicios o recursos) y el servidor (que proporciona esos servicios o recursos). ⁴⁰



Arquitectura Cliente-Servidor .40.

Arquitecturas limpias

Las arquitecturas limpias, hexagonales y en capas son enfoques similares que dividen un sistema en capas con responsabilidades específicas y comunicación controlada entre ellas. ⁴¹



Arquitecturas Limpias .41.

40. Dos Setenta. (s.f.). ¿Qué es una api rest?. Recuperado de <https://dossetenta.com/que-es-una-api-rest/>

41. Medium. (s.f.). Apuntes sobre Arquitecturas Limpias (de Software)— Vol 1. Recuperado de <https://medium.com/@PabloLeonPsi/apuntes-sobre-arquitecturas-limpas-de-software-vol-1-1a91131933e2>



...

Desarrollo Móvil



SPLPC™ Versión 032025





Imagen de Thom Bradley en Unsplash

Desarrollo Móvil

"El desarrollo de aplicaciones móviles es el proceso de creación de software para smartphones, tablets y asistentes digitales, más comúnmente para los sistemas operativos Android e iOS".

42. IBM. (s.f.). ¿Qué es el desarrollo de aplicaciones móviles? IBM. Recuperado de <https://www.ibm.com/mx-es/topics/mobile-application-development>





Tipos de Aplicaciones Móviles

1

Aplicaciones Nativas

Son aplicaciones móviles desarrolladas específicamente para un sistema operativo en particular, lo que permite un rendimiento óptimo y una experiencia de usuario altamente personalizada en dispositivos móviles.

2

Tecnologías

Los sistemas operativos más comunes son iOS y Android. Utilizan lenguajes como Swift, Objective-C, Java y Kotlin.

3

IDE

Entorno de Desarrollo Integrado como Xcode para iOS y Android Studio para Android.



43. Design. (2022). **Principales tipos de aplicaciones y ejemplos por sectores** Recuperado de: <https://appdesign.dev/principales-tipos-de-aplicaciones-y-ejemplos-por-sectores/>



Tipos de Aplicaciones Móviles

Aplicaciones Multiplataforma

Son aplicaciones diseñados para funcionar en diferentes sistemas operativos y plataformas, como iOS, Android, y a menudo también en web y escritorio, utilizando un conjunto común de tecnologías y código base. 44

Híbridas

Se desarrollan utilizando tecnologías web estándar, como HTML, CSS y JavaScript, y luego se envuelven en un contenedor nativo que permite su ejecución en diferentes plataformas, como iOS y Android.

Tecnologías

Frameworks

Lenguajes



Javascript

PWA

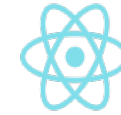
Apps Web Progresivas

Son aplicaciones web que ofrecen una experiencia similar a la de las aplicaciones nativas, pero se ejecutan en navegadores web estándar sin necesidad de instalación una tienda móvil.

Tecnologías Frameworks



Angular



React



Vue

44. EMMA Arkana (s. f.). **Tipos de aplicaciones, características, ejemplos y comparativa** Recuperado de <https://emma.io/blog/tipos-aplicaciones-caracteristicas-ejemplo>





Cloud



SPLPC™ Versión 032025



Cloud

Computación en la nube ⁴⁵

Modelo de entrega de servicios informáticos en el que los recursos se proporcionan a los usuarios a través de Internet.

En esta modalidad, el cliente alquila los equipos informáticos a un **proveedor** de servicios Cloud, quien se encargará de todo el mantenimiento y sus actualizaciones.



Imagen de [Freepik](#)

¿Cuándo utilizarla?

- ✓ Escalabilidad
- ✓ Menores costos iniciales
- ✓ Actualizaciones automáticas

Tener en cuenta

- ✓ Dependencia en conectividad
- ✓ Control del proveedor
- ✓ Costos a largo plazo

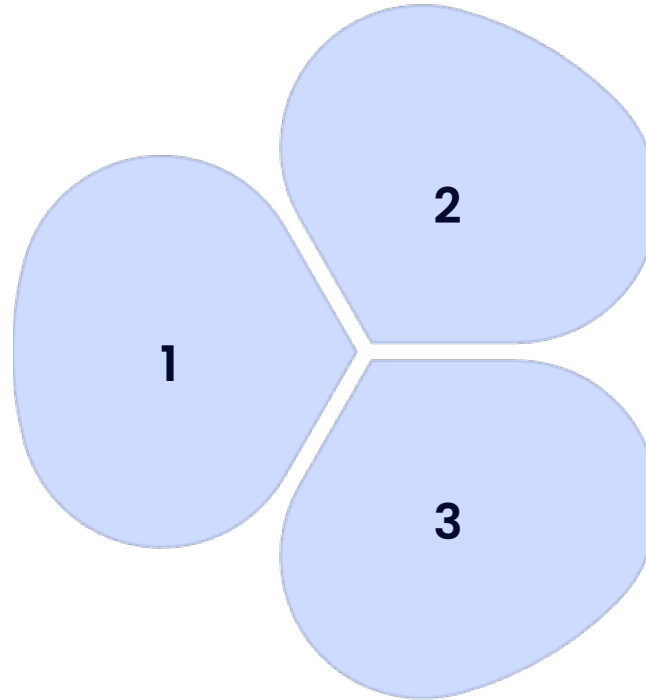
45. Atlassian. (s.f) ¿Qué es la computación en la nube? Recuperado de <https://www.atlassian.com/es/microservices/cloud-computing>



Tipos de nube

Nube pública

Un entorno de servicios en la nube que ofrece recursos computacionales, almacenamiento y servicios gestionados de manera **compartida** a través de Internet.



Nube privada

Está compuesta por recursos informáticos en la nube que utiliza **exclusivamente una organización**. En una nube privada, los servicios y la infraestructura siempre se mantienen en una red privada, y el **hardware y software se dedican únicamente a una organización**.

Nube híbrida

Un enfoque que combina entornos de nube pública y privada, permitiendo la transferencia de datos y aplicaciones entre ellos. Proporciona flexibilidad para las organizaciones al equilibrar la escalabilidad y la eficiencia de la nube pública con el control y la personalización de la nube privada.



Tipos de servicios en la nube

IaaS – Infraestructura como servicio

Proporciona la infraestructura básica de TI, como servidores, almacenamiento y redes. El cliente es responsable de instalar y gestionar el software y las aplicaciones.

Ejemplos: Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3)

SaaS – Software como servicio

Proporciona software de aplicaciones listo para usar que se ejecuta en la nube. El cliente no tiene que preocuparse por la instalación, la gestión o la actualización del software.

Ejemplos: Microsoft 365 (anteriormente Office 365), Salesforce

1

2

PaaS – Plataforma como servicio

Proporciona una plataforma completa para desarrollar, implementar y gestionar aplicaciones. El cliente es responsable de desarrollar e implementar las aplicaciones.

Ejemplos: Firebase, Amplify (AWS)

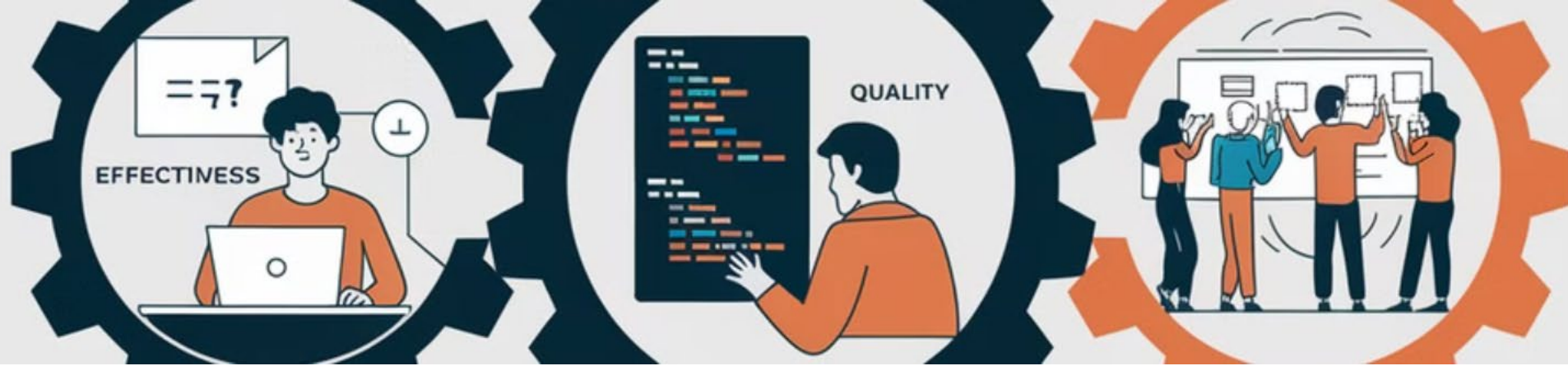
3

*Si bien existen otros servicios en la nube, en esta presentación se cubren los tres modelos más comunes.



Habilidades Técnicas





Habilidades Técnicas

3

Pilares esenciales

Efectividad, calidad y equipo: los fundamentos del liderazgo técnico

1

Objetivo principal

Maximizar la entrega de valor mediante habilidades técnicas bien desarrolladas

El desarrollo de software constantemente se ve enfrentado a cómo aumentar la puesta en producción de funcionalidades (**efectividad**) y la confianza en ellas (**calidad**), esto en torno a un grupo de profesionales motivados (**equipo**).

Estos tres retos configuran las habilidades técnicas asociadas al líder de proyectos de software.



...

Efectividad



SPLPC™ Versión 032025



Efectividad

La efectividad para los líderes de proyectos de software se refiere a su **capacidad para lograr los resultados deseados de manera satisfactoria**, a lo largo del ciclo de vida del desarrollo de software.

Elementos claves

- Output
- Outcome
- 4 elementos

Elementos claves

- Time to Market
- Lead Time
- Cycle Time
- Throughput / Rendimiento



Elementos claves





Output

Definición

Se refiere a los productos o servicios tangibles que se generan como resultado de un proceso o actividad.

Lo que se produce

- Un informe final
- Un nuevo software
- Una nueva funcionalidad
- Un evento realizado

Ejemplos



El nuevo sistema de gestión de proyectos está instalado y configurado

Los usuarios han recibido capacitación sobre cómo usar el nuevo sistema



La infraestructura de TI se ha migrado a la nube.

Los usuarios tienen acceso a sus aplicaciones y datos en la nube.





Outcome

1 Definición

Se refiere al impacto o efecto que se produce como resultado de un output. Es el cambio positivo que se busca lograr a través de un proyecto o iniciativa.

2 El impacto que se logra

Aumento de la satisfacción del cliente, mejora en la eficiencia operativa, reducción de costos, mayor participación en el mercado.

3 Ejemplos



Mejora en la agilidad y escalabilidad de la infraestructura de TI: Mayor facilidad para escalar la infraestructura en función de la demanda.



Mayor seguridad de la información: Protección de los datos contra amenazas cibernéticas.



Objetivos y Recursos



Cumplimiento de Objetivos

La efectividad implica lograr los objetivos del proyecto, que pueden incluir la entrega de un producto de software funcional de calidad, dentro de los plazos acordados.

Es indispensable, el establecimiento de objetivos claros, alcanzables y medibles, que se encuentren en constante conversación.



Gestión de Recursos

Los líderes de proyectos de software deben ser efectivos en la gestión del presupuesto y recursos, esto incluye: herramientas, proveedores, infraestructura, licenciamientos, etc.



Riesgos e interesados



Gestión de Riesgos

La identificación y acción oportuna de riesgos potenciales (tanto positivos como negativos), favorece el cumplimiento de objetivos.

Una matriz de riesgos (liviana) en constante trabajo y con acciones concretas aumentará la confianza en los resultados.



Interesados

Los líderes de proyectos deben ser efectivos en la comunicación con su equipo y otras partes interesadas.

La colaboración efectiva, favorece la comprensión de necesidades y expectativas del cliente, el progreso y solución de problemas de forma oportuna.

Hacer uso de herramientas de gestión que permitan visibilizar el trabajo, identificar mejoras, impedimentos y necesidades es esencial.





Métricas



T2M – Time to Market

Definición	Tiempo promedio transcurrido entre el momento en que se concibe una idea y el momento en que se lanza un producto, también conocido como Tiempo de comercialización .
Objetivo de medirlo	Garantizar la entrega oportuna, midiendo la velocidad en la que el producto llega al mercado, con el objetivo de mejorar posicionamiento, la competitividad, y los niveles de satisfacción.
Aplicación	<p>"Time to Market " se utiliza para:</p> <ul style="list-style-type: none">• Aprender y adaptarse más rápido con base a la retroalimentación del mercado y los usuarios.• Maximizar la eficiencia y la competitividad de un proyecto (Eric Ries)• Minimizar el riesgo de desarrollar productos que no tengan demanda real en el mercado.
Ejemplo	<p>Fecha iniciativa: 5 Junio Lanzamiento MVP: 5 Octubre</p> <p>T2M: Desde la fecha iniciativa hasta la fecha entrega a comercialización T2M: Desde el 5 Junio al 5 Octubre (Normalmente se usan meses) T2M: 4 Meses</p>



Lead Time / Tiempo de espera

Definición	Es el tiempo que transcurre desde el momento en que se inicia una solicitud hasta el momento en que se completa y se entrega al cliente o usuario final.
Objetivo de medirlo	Entender el tiempo de cuanto lleva completar una tarea, proceso o proyecto desde su inicio, lo que permite identificar posibles retrasos y oportunidades de mejora
Aplicación	<p>“Lead Time” se utiliza para:</p> <ul style="list-style-type: none">• Evaluar la eficiencia del proceso de desarrollo, desde la concepción de una característica hasta su entrega.• Identificar cuellos de botella y obstáculos en el flujo de trabajo.• Determinar el tiempo necesario para cumplir con los plazos de entrega
Ejemplo	<p>Inicio de Trabajo : El equipo de desarrollo comienza a trabajar en la solicitud el 1 de enero. Fecha de lanzamiento mercado: 15 de marzo.</p> <p>LT : Desde el 1 Enero al 15 de Marzo LT : 2 meses y 14 días</p>



Cycle Time / Tiempo de Ciclo

1

Definición

El Cycle Time es el tiempo que transcurre desde que se comienza a trabajar en una tarea o elemento de trabajo (como una característica, tarea, historia de usuario, o cualquier unidad de trabajo) hasta que se completa y entrega (cumple con la definición de terminado).

2

Objetivo de medirlo

Analizar el tiempo real que se necesita para completar cada unidad de trabajo, para planificar los plazos y ajustar la capacidad del equipo.

3

Aplicación

"Cycle Time" se utiliza para:

- Evaluar el tiempo real que toma el ciclo de desarrollo
- Identificar ineficiencias en el proceso y cuellos de botella.

Ejemplo

Inicio de Trabajo : El equipo de desarrollo comienza a trabajar en la solicitud el 2 de enero.

Fecha de terminación desarrollo : 31 de enero

CT : Desde el 2 enero al 31 de enero

CT : 29 días



Cycle Time / Tiempo de Ciclo ⁵⁰

Definición	El Cycle Time es el tiempo que transcurre desde que se comienza a trabajar en una tarea o elemento de trabajo (como una característica, tarea, historia de usuario, o cualquier unidad de trabajo) hasta que se completa y entrega (cumple con la definición de terminado).
Objetivo de medirlo	Analizar el tiempo real que se necesita para completar cada unidad de trabajo, para planificar los plazos y ajustar la capacidad del equipo.
Aplicación	<p>"Cycle Time" se utiliza para:</p> <ul style="list-style-type: none">• Evaluar el tiempo real que toma el ciclo de desarrollo• Identificar ineficiencias en el proceso y cuellos de botella.
Ejemplo	<p>Inicio de Trabajo : El equipo de desarrollo comienza a trabajar en la solicitud el 2 de enero. Fecha de terminación desarrollo : 31 de enero</p> <p>CT : Desde el 2 enero al 31 de enero CT : 29 días</p>

50. Sentries. (2022, abril 19). Cycle Time: qué es, cómo medirlo y mejorarlo en desarrollo de software. Sentries. <https://sentries.io/blog/cycle-time-que-es/>



Throughput / rendimiento

Definición	El Throughput corresponde a el número de elementos completados y que se entregan en una unidad de tiempo.
Objetivo de medirlo	Evaluar la capacidad de producción de un sistema, equipo o proceso en un período de tiempo determinado.
Aplicación	<p>"Throughput" se utiliza para:</p> <ul style="list-style-type: none">• Identificar áreas donde se pueden realizar mejoras para aumentar la entrega.• Identificar cuellos de botella y limitaciones en el flujo de trabajo.
Ejemplo	<p>Tiempo de Trabajo : 2 semanas Trabajo comprometido: 12 solicitudes</p> <p>Tp : 12 solicitudes para cada 2 semanas (Bloque de tiempo previamente establecido)</p>



Pautas Claves

1 Velar por la mejora continua

Cada aspecto a lo largo del tiempo es susceptible de mejora. Establecer ciclos cortos de trabajo (≤ 1 mes), favorecen la evaluación y acción para aumentar la efectividad.

2 Time to Market

Mejorar el **Time to Market** puede conducir a una **mayor satisfacción del cliente** y a una posición más competitiva en la industria.

3 Lead Time

Se puede utilizar el **Lead Time** para **identificar oportunidades de mejora**, ajustar plazos de entrega y asegurarse de que los proyectos se completen de manera eficiente.

4 Throughput

El **Throughput** proporciona información sobre la eficiencia del equipo y puede utilizarse para identificar áreas de mejora en el proceso de desarrollo **y la gestión de la carga de trabajo**.

5 Cycle Time

Con el **Cycle Time**, el equipo de desarrollo puede identificar oportunidades para acelerar el flujo de trabajo, **reducir los tiempos de entrega y mejorar la productividad**.



Calidad





Calidad

Para los líderes de proyectos de software, la calidad se refiere a la medida en que el software cumple con los estándares y requisitos establecidos, y satisface las necesidades y expectativas del cliente y los usuarios finales.

Elementos claves

- Atributos de calidad del Software
- Pruebas de calidad

Métricas

- Nivel de aceptación de pruebas funcionales
- Defectos Activos
- Pruebas Automatizadas
- Cobertura de Código



Elementos claves



Atributos de calidad del software



Un atributo de calidad es una **propiedad medible de un sistema**, asociado al cumplimiento de las necesidades del mismo.

En ocasiones se refieren a requisitos no funcionales, sin embargo, en este planteamiento se consideran los requisitos funcionales y la conformidad con ellos, la base inicial de la calidad.

Adicionalmente, existen un sin-número de características que pueden hacer de un sistema un elemento sólido, aquí se destacan algunos atributos clave en la industria del software.



Pruebas de calidad



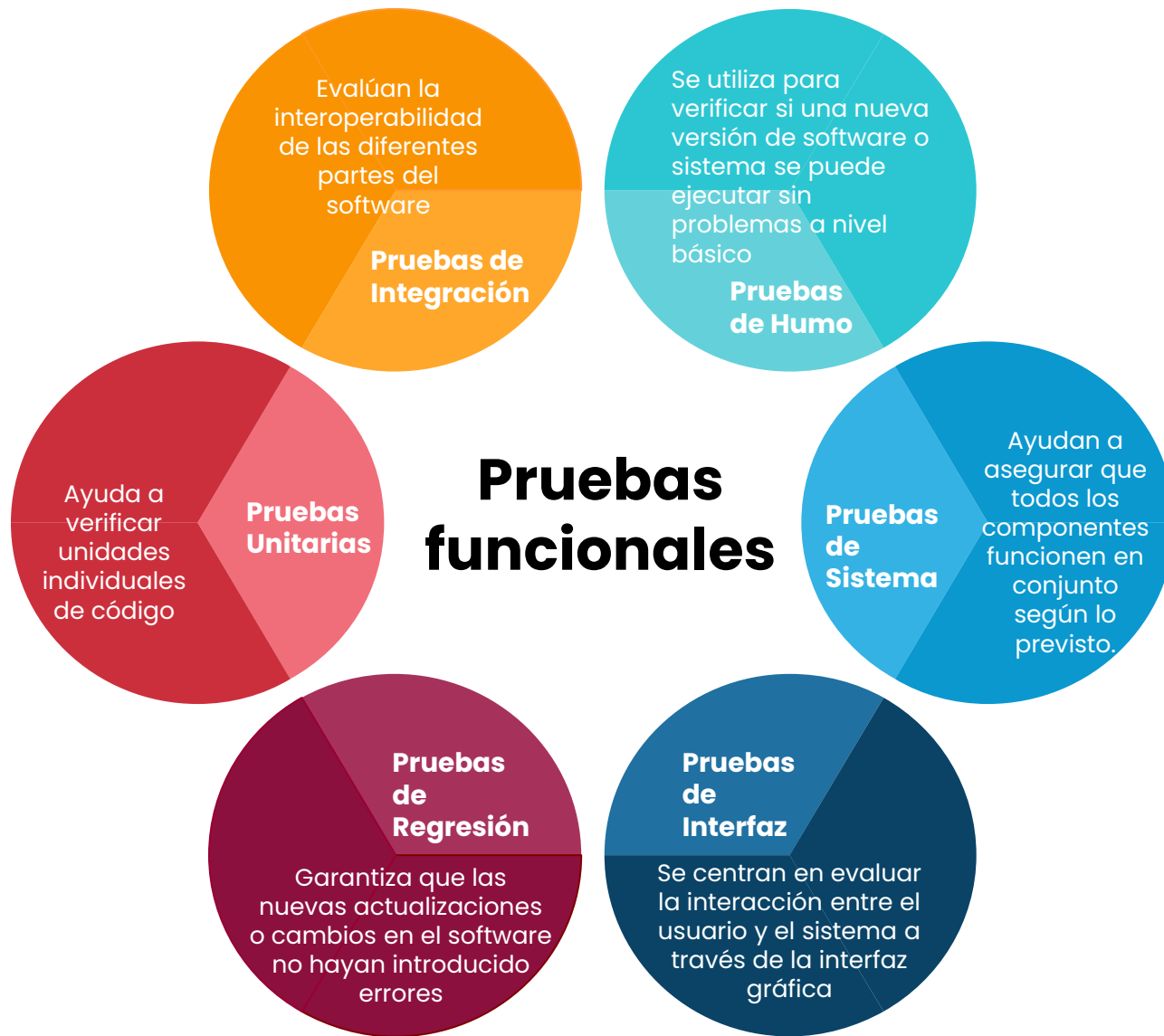
Las pruebas de software son el principal aliado de la calidad.

Permite:

- Prevención de errores y reprocesos
- Reducción de costos de desarrollo
- Mejora del rendimiento
- Favorece la satisfacción del usuario y su confianza

De acuerdo con su objeto, de si se **intenta evaluar un aspecto concreto de la operación del sistema** (funcional) o un elemento transversal (No funcional), existen diferentes categorías.





Son un tipo de evaluación que se enfoca en verificar si un software cumple con los requisitos y especificaciones establecidas para su correcto funcionamiento.

Estas pruebas evalúan las características y comportamientos del software para garantizar que se comporta de acuerdo a lo esperado.

Las pruebas funcionales se basan en la comprensión de los requisitos y especificaciones del software.





Se centran en evaluar elementos que no están relacionados directamente con la funcionalidad del sistema.

Estos elementos incluyen el **rendimiento, la usabilidad, la escalabilidad, la seguridad** y otros atributos que afectan la experiencia del usuario.

Las pruebas no funcionales se basan en el comportamiento del sistema en condiciones específicas.



Pirámide de Cohn ⁴⁶



Enfoques de desarrollo

TDD (Desarrollo guiado por Pruebas) Pruebas Unitarias

BDD (Desarrollo guiado por Comportamiento) Pruebas Funcionales

Análisis código: Revisión código estático
Tecnología: SonarQube

⁴⁶<https://aicsvirtual.org/modelo-de-la-piramide-de-automatizacion-para-pruebas-de-performance/>



Métricas

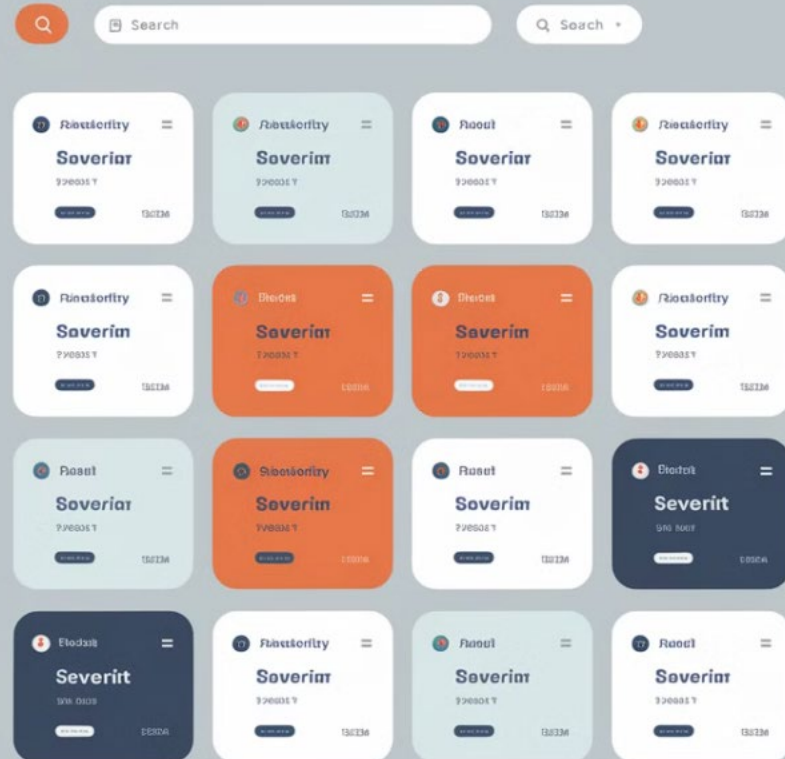


Nivel de aceptación de pruebas funcionales

Definición	Es la medida que indica qué porcentaje de las pruebas funcionales realizadas ha pasado con éxito respecto a los criterios de aceptación definidos de cada funcionalidad.
Objetivo de medirlo	Determinar si un sistema de software cumple con los requisitos funcionales y las expectativas de los usuarios finales antes de su lanzamiento.
Aplicación	Se aplica al final del ciclo de pruebas de software, generalmente después de las pruebas unitarias, de integración y de sistema. Implica la ejecución de pruebas funcionales por parte de usuarios o stakeholders que evalúan si el software cumple con sus necesidades y requisitos específicos. Si el software supera estas pruebas de aceptación, se considera listo para ser desplegado en un entorno de producción.
Ejemplo	<p># escenarios de prueba definidos: 100 # escenarios de prueba satisfactorios: 90</p> <p>NAPF: $(\# \text{ escenarios de prueba satisfactorios} / \text{escenarios de prueba definidos}) * 100$ NAPF : $(90/100) * 100$ NAPF : 90%</p>



Bug Tracking Software Defects



Defectos Activos

Definición

Representa la cantidad de defectos o problemas no resueltos en un software o sistema en un momento específico del ciclo de desarrollo.

Objetivo de medirlo

Mantener un registro de los problemas no resueltos en el software y evaluar el nivel de calidad y estabilidad del sistema e identificar áreas críticas de mejora y priorizar la asignación de recursos para abordar estos problemas.

Aplicación

Se pueden utilizar para tomar decisiones sobre la liberación de una versión, la asignación de recursos para correcciones de errores o para identificar patrones y tendencias en la calidad del software.

Ejemplo

Defectos reportados: 25

Defectos resueltos: 3

DA: # Defectos reportados - # Defectos resueltos

DA : 25-3

DA : 22



Pruebas Automatizadas



Definición

Son un conjunto de técnicas y procesos que utilizan herramientas y scripts para ejecutar casos de prueba de forma automatizada. Estos casos de prueba están diseñados para verificar el comportamiento y la funcionalidad del software de manera consistente y repetible.



Objetivo de medirlo

Mejorar la eficiencia y la efectividad del proceso de pruebas al reducir el tiempo y los recursos necesarios para ejecutar pruebas repetitivas, permitiendo así a los equipos de desarrollo y QA centrados en áreas de mayor complejidad y en pruebas exploratorias.



Aplicación

Se aplican en pruebas unitarias, pruebas de integración, pruebas de regresión y pruebas de carga y rendimiento. Se utilizan para verificar que el software cumple con los requisitos y las expectativas del cliente de manera rápida y confiable.



Ejemplo

Escenarios de pruebas automatizadas: 100

Escenarios de pruebas fallidas: 5

PA: (Escenarios de pruebas automatizadas - # Escenarios de pruebas fallidos) / Escenarios de pruebas automatizadas * 100

PA: $((100 - 5) / 100) * 100$

PA: 95%



Cobertura de Código

Definición	Se refiere al porcentaje de líneas de código que se definen para la ejecución de un conjunto de pruebas con respecto al total de líneas de código desarrolladas.
Objetivo de medirlo	Evaluar la efectividad de las pruebas realizadas y determinar qué parte del código ha sido validado.
Aplicación	Puede ser utilizado en diferentes niveles de pruebas, incluyendo pruebas unitarias, pruebas de integración y pruebas de sistema. También es útil para identificar áreas del código que no han sido probados, lo que puede indicar la necesidad de casos de prueba adicionales.
Ejemplo	<p># Líneas de código desarrolladas : 10.000</p> <p># Líneas de código para revisión: 8.000</p> <p>CC: (Líneas de código para revisión/ # Líneas de código desarrolladas) *100</p> <p>CC : (8.000/10.000)*100</p> <p>CC : 80%</p>



Pautas claves



Pautas Claves

1 Nivel de aceptación de pruebas funcionales

Permite a las organizaciones asegurarse de que el software sea lo suficientemente estable, minimizando los riesgos de fallos graves y problemas de usabilidad. Se busca alcanzar un alto nivel de aceptación, pero esto puede variar según las condiciones específicas de cada proyecto.

2 Defectos Activos

La identificación de los Defectos Activos permite a los equipos tener una visión clara de los desafíos pendientes y tomar decisiones informadas sobre cuándo y cómo abordar la resolución de los defectos. Cuanto menor sea el número de Defectos Activos, mejor será la calidad percibida del software.

3 Pruebas automatizadas

Es importante tener en cuenta que un alto porcentaje de éxito en las pruebas automatizadas no garantiza la ausencia de problemas, pero puede indicar un buen nivel de confiabilidad en el código y la funcionalidad probados. También es esencial mantener y actualizar periódicamente las pruebas automáticas para mantener su relevancia y precisión a medida que el software evoluciona.

4 Cobertura de Código

Tener en cuenta que una Cobertura de Código alta no garantiza que no haya errores, pero sugiere que se ha llevado a cabo un buen esfuerzo para validar el código. Los equipos de desarrollo a menudo buscan lograr una cobertura lo más cercana posible al 100%, aunque esto puede variar según el contexto del proyecto.



Equipo



Equipo de desarrollo

Es un grupo de profesionales que trabajan en conjunto para planificar, ejecutar todas las actividades relacionadas con el diseño, desarrollo y continuidad de una aplicación de software.

En su elección se busca generar confianza e ir hacia la consolidación del alto desempeño.

Elementos claves

- Equipos base
- Equipos ampliados

Métricas

- Índice moral del equipo ó métrica de felicidad
- Confiabilidad en compromiso
- Cumplimiento de Trabajo en progreso/ Work In Progress (WIP)



Elementos claves



Equipos Base



Desarrollador/a Back-end

Crea y administra la parte no visible de una aplicación web.

Su trabajo se centra en la gestión de servidores, bases de datos, lógica de negocio y la infraestructura que permite que la aplicación funcione sin problemas.



Desarrollador/a Front-end

Codifica y optimiza la interfaz de usuario de aplicaciones web. Es encargado de crear una experiencia de usuario atractiva y funcional en la parte visible de una aplicación.



Quality Assurance (QA)

Responsable de evaluar y garantizar la calidad de un producto de software mediante la identificación y resolución de problemas, pruebas exhaustivas y aseguramiento de estándares y requisitos previamente definidos.



Arquitecta/o de Software

Diseña la estructura y la organización de un sistema de software, definiendo la interacción entre componentes y módulos. Toma decisiones sobre la tecnología, la arquitectura y las mejores prácticas de desarrollo para garantizar que el software sea escalable, eficiente y cumpla con los requisitos del proyecto

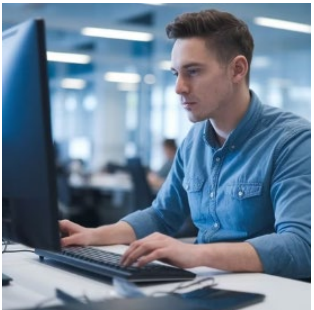


Equipos ampliados



Diseñador/a UX/UI

Diseña la interfaz de usuario de aplicaciones web, centrándose en la experiencia del usuario para lograr una navegación intuitiva y atractiva.



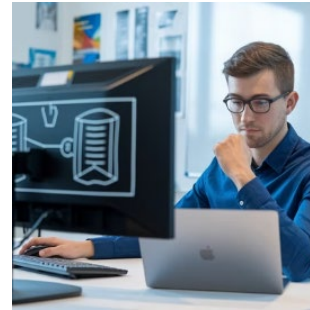
QA Automation

Es el líder de la automatización de pruebas de software para garantizar la calidad y fiabilidad de las aplicaciones, acelerando el proceso de pruebas y aumentando la eficiencia.



Prof. Seguridad Información

Es el responsable de desarrollar protocolos de gestión de riesgos para evitar incidentes y amenazas contra la información.



DBA

Responsable de gestionar y mantener bases de datos, asegurando su rendimiento, seguridad y disponibilidad.



Ing. Cloud

Es el responsable de la gestión, implementación y optimización de soluciones en la nube, como AWS, Azure o Google Cloud, para alojar aplicaciones y servicios de manera eficiente y segura.



Ing. CI/CD

Automatiza y optimiza el flujo de desarrollo de software, desde la integración de código hasta su entrega final, garantizando una entrega rápida y confiable de nuevas funcionalidades y correcciones.



Los siguientes son algunos roles fundamentales que se encuentran en el **front-end**:

- ✓ **Diseñador/a UI/UX:** se encarga de crear la apariencia visual (UI) y la experiencia del usuario (UX) en una aplicación web. Diseña la interfaz gráfica para garantizar que sea atractiva y fácil de usar para los usuarios.
- ✓ **Maquetador/a (HTML y CSS):** se dedica a convertir los diseños en código HTML y CSS. Crea la estructura y el diseño visual de las páginas web para que se vean según lo planeado. Es posible que este rol se asuma por parte del diseñador/a o el/la desarrollador/a.
- ✓ **Desarrollador/a Front-end:** Se asegura de que los elementos web respondan a las acciones del usuario (interactividad) y funcionen de manera fluida.
- ✓ **Analista de Calidad (Quality Assurance - QA) para Front-end:** Se encarga de probar y garantizar la calidad de la interfaz y la experiencia del usuario en una aplicación web. Realiza pruebas para identificar defectos relacionados con la apariencia y el comportamiento de la interfaz.

47. Valdivia-Caballero, J. J. (2016). Modelo de procesos para el desarrollo del front-end de aplicaciones web. *Interfases*, (009), 187-208. Recuperado de <https://www.researchgate.net/publication/320914166>



Los siguientes son algunos roles comunes que podemos encontrar en el **back-end** :

- ✓ **Desarrollador/a Back-end:** se encarga de crear y mantener la lógica de negocio de una aplicación web. Su responsabilidad es gestionar el procesamiento de datos en el servidor y garantizar que las solicitudes del front-end se manejen de manera efectiva.
- ✓ **Analista de Calidad (Quality Assurance - QA) para Backend:** se enfoca en probar y garantizar la calidad de los servicios web y funcionalidades específicas del back-end de una aplicación. Realiza pruebas de integración, carga, rendimiento, etc y se asegura de que los datos se procesen correctamente.

En el ámbito del desarrollo back-end, existen numerosos roles especializados según las necesidades específicas de un proyecto. Estos roles pueden incluir especialistas en bases de datos, desarrolladores de APIs, arquitectos de software, entre otros.

48. Servnet. (2021). Backend y Frontend, ¿Qué es y cómo funcionan en la programación? Recuperado de https://www.researchgate.net/publication/320914166_Modelo_de_procesos_para_el_desarrollo_del_front-end_de_aplicaciones_web



Roles

Los siguientes son algunos roles comunes que podemos encontrar en **bases de datos**:

- ✓ **Administrador/a de Bases de Datos (DBA):** Responsable de diseñar, implementar y gestionar bases de datos. Administra el rendimiento, la seguridad y la disponibilidad de los sistemas de bases de datos.
- ✓ **Desarrollador/a de bases de datos:** Responsable de diseñar, desarrollar y mantener la estructura y la lógica de la base de datos. Este rol se enfoca en la creación de tablas, consultas, procedimientos almacenados y otras componentes de la base de datos. Garantiza que los datos se almacenen y recuperen eficientemente, además de optimizar el rendimiento del sistema de bases de datos. También trabaja en la integración de la base de datos con aplicaciones y sistemas para asegurar un flujo de datos efectivo.
- ✓ **Analista de datos:** Se enfoca en la interpretación y el análisis de los datos almacenados en la base de datos, proporcionando información valiosa para la toma de decisiones.





Métricas



Índice moral del equipo o métrica de felicidad

Definición	Evaluación del nivel de satisfacción y motivación en el entorno laboral de los miembros del equipo. La periodicidad la define el líder.
Objetivo de medirlo	Identificar áreas de mejora en el ambiente para promover un entorno laboral positivo que fomente la satisfacción y la motivación.
Aplicación	Identifica problemas potenciales que podrían afectar la productividad y la colaboración. Su instrumento es una valoración de satisfacción y bienestar en un rango de 1 a 5, donde 1 es muy insatisfecho y 5 muy satisfecho.
Ejemplo	Rango de calificación: 1-5, siendo 1 muy insatisfecho y 5 Muy satisfecho Calificación de cada miembro del equipo: C# IM: Sumatoria todas las calificaciones de los miembros del equipo IM: $(C1+C2+C3+C4+C5)\#personas$ del equipo que evaluaron IM: $(5+4+4+3+5)5$ IM: 4,2



Confiabilidad en compromiso



Definición

Es el ratio entre los compromisos (puntos, funcionalidades) adquiridos y los compromisos entregados (puntos, funcionalidades) por el equipo. La razón entre lo comprometido / lo realizado.



Objetivo de medirlo

Evaluar la consistencia de confiabilidad de un equipo frente a sus compromisos, identificando patrones de cumplimiento.



Aplicación

Puede utilizarse para identificar áreas de mejora en la gestión del tiempo y la comunicación, y para tomar decisiones informadas sobre la asignación de responsabilidades, evaluando la puntualidad y la consistencia en el cumplimiento de plazos y metas.

Ejemplo

Compromisos adquiridos: 14

Compromisos entregados : 11

CC: $(\# \text{Compromisos entregados} / \# \text{Compromisos adquiridos}) * 100$

CC : $(11/14) * 100$

CC : 78%



Confiabilidad en compromiso

Definición	Es el ratio entre los compromisos (puntos, funcionalidades) adquiridos y los compromisos entregados (puntos, funcionalidades) por el equipo. La razón entre lo comprometido / lo realizado.
Objetivo de medirlo	Evaluar la consistencia de confiabilidad de un equipo frente a sus compromisos, identificando patrones de cumplimiento.
Aplicación	Puede utilizarse para identificar áreas de mejora en la gestión del tiempo y la comunicación, y para tomar decisiones informadas sobre la asignación de responsabilidades, evaluando la puntualidad y la consistencia en el cumplimiento de plazos y metas.
Ejemplo	<p># Compromisos adquiridos: 14 # Compromisos entregados : 11</p> <p>CC: ($\# \text{Compromisos entregados} / \# \text{Compromisos adquiridos}$)*100 CC : $(11/14)*100$ CC : 78%</p>



Cumplimiento de Trabajo en progreso/ Work In Progress (WIP) ⁴⁹

Definición	El Work In Progress se refiere a las tareas o elementos que han sido iniciadas, pero aún no han sido completadas o entregadas.
Objetivo de medirlo	Es evaluar el estado del flujo de trabajo y la capacidad del sistema. Optimizando el trabajo en curso y la capacidad para completar tareas de manera efectiva. Ayuda a el enfoque y control del estrés del equipo.
Aplicación	Utilizado para evaluar la carga de trabajo actual del equipo y garantizar el cierre de los ítems en progreso e identificar tareas en progreso durante demasiado tiempo que pueden requerir atención pronta.
Ejemplo	<p># ítems en progreso: 6 Máximo # ítems en progreso: 4</p> <p>WIP: $(\# \text{ ítems en progreso} / \text{Máximo } \# \text{ ítems en progreso}) * 100$ WIP : $(6/4)*100$ WIP : 66%</p>

49. Molina, D. (2022). Qué es Work in Progress (WIP) en Kanban y cómo utilizarlo. Thinking for Innovation. <https://www.iebschool.com/blog/que-es-work-in-progress-wip-en-kanban-y-como-utilizarlo-agile-scrum/>



Pautas claves



Pautas clave

- Conocer al equipo (Identificar **roles y talentos**).
- Motivar al equipo con **objetivos claros y retroalimentación constante**.
- Promover la **colaboración y autogestión** en todos los miembros del equipo.
- Mantener un **WIP** equilibrado puede ayudar al equipo a enfocarse en completar tareas antes de iniciar nuevas, lo que a su vez puede aumentar la productividad, la calidad del trabajo y disminuye el estrés del equipo.
- Monitorear el **Índice Moral del Equipo** ayuda a identificar áreas de mejora, problemas potenciales, motivaciones del equipo para favorecer un entorno de trabajo saludable y una cultura empresarial positiva.
- Es importante revisar los patrones de cumplimiento para ajustar la capacidad del equipo en el análisis de la **Confiabilidad en Compromiso**.



Referencias Bibliográficas

- Abi.gitbook.io. (s.f.). Servicios REST. [online]. Recuperado de <https://abi.gitbook.io/net-core/3.-servicios-rest/3.1-servicios-rest>
- Acens for Telefónica Company (2015). Whitepaper: Control de Versiones GIT y GITHUB. Recuperado de <https://www.acens.com/comunicacion/white-papers/control-versiones-git-github/>
- Aizprua, S., Ortega, A., & Von Chong, L. (2019). Calidad del Software una Perspectiva Continua. Centros: Revista Científica Universitaria, 8(2), 120-134. Recuperado de <http://portal.amelica.org/ameli/jatsRepo/228/228986011/html/index.html>
- Amazon Web Services. (s.f.). ¿Qué son los estándares de SQL? [en línea]. Recuperado de <https://aws.amazon.com/es/what-is/sql/>
- Amazon Web Services. (s.f.). Aspectos importantes de las bases de datos relacionales [en línea]. Recuperado de <https://aws.amazon.com/es/relational-database/>
- Ariza, D. A. (2017). Efectividad de la gestión de los proyectos: una perspectiva constructivista. Obras y proyectos, (22), 75-85. Recuperado de https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-28132017000200075
- Atlassian (s.f.). ¿Qué es el Control de Versiones? Recuperado de <https://www.atlassian.com/es/git/tutorials/what-is-version-control>
- Atlassian Bitbucket (s.f.). Software de control de versiones para equipos profesionales. Recuperado de <https://bitbucket.org/product/es/version-control-software>
- AWS (s.f.). ¿Qué es el ciclo de vida del desarrollo de software (SDLC)? Recuperado de <https://aws.amazon.com/es/what-is/sdlc/>
- AWS. (s.f.). ¿Qué es un API? AWS. [online]. Recuperado de <https://www.oracle.com/database/what-is-database.html>
- Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice. Addison-Wesley
- Canalceo (2019). 10 lecciones (aún vigentes) del pensamiento de Peter Drucker Recuperado de <https://canalceo.theobjective.com/10-lecciones-aun-vigentes-del-pensamiento-de-peter-drucker/>
- Caridad, M., Batista, J., & Castellano, M. I. (2007). Comunicación revisitada en el contexto de las organizaciones empresariales. Clío América, 1(2), 240-258. Recuperado de <https://dialnet.unirioja.es/descarga/articulo/5114837.pdf>



Referencias Bibliográficas

- Catenon World (s.f.). 7 Roles en Software que Debes Tener en Cuenta – y cómo distinguirlos Recuperado de <https://blog.catenon.com/es/roles-software-atraer/>
- Covey, S. R. (2015). Los 7 Hábitos de la Gente Altamente Efectiva: Edición de Imágenes. Mango Media Inc. Recuperado de <http://200.70.33.130/images2/SGHPE/7%20HABITOS.pdf>
- Design (2022). Principales tipos de aplicaciones y ejemplos por sectores Recuperado de: <https://appdesign.dev/principales-tipos-de-aplicaciones-y-ejemplos-por-sectores/>
- Digital IONOS (s.f) ¿Qué es CSS? Definición y aplicación. Recuperado de <https://www.ionos.es/digitalguide/paginas-web/disenio-web/que-es-css/>
- Dos Setenta. (s.f.). ¿Qué es una api rest?. Recuperado de <https://dossetenta.com/que-es-una-api-rest/>
- EMMA Arkana (s. f.) Tipos de aplicaciones, características, ejemplos y comparativa Recuperado de <https://emma.io/blog/tipos-aplicaciones-caracteristicas-ejemplos/>
- Forbes (2021). El Candor Radical: de habilidades blandas a esenciales. Recuperado de <https://www.forbes.com.mx/red-forbes-el-candor-radical-de-habilidades-blandas-a-esenciales/>
- GeeksforGeeks. (2023)– 10 Best Web Development Frameworks to Use in 2023.. Recuperado de <https://hackr.io/blog/web-development-frameworks>
- GeeksforGeeks. (2023). Top Front-End Frameworks in 2023. [online]. Recuperado de <https://www.geeksforgeeks.org/top-front-end-frameworks/>
- GeeksforGeeks. (2023). Web Development – GeeksforGeeks. [online]. Recuperado de <https://www.geeksforgeeks.org/web-development/>
- IBM. (s.f.). ¿Qué es el desarrollo de aplicaciones móviles? IBM. [online]. Recuperado de <https://www.ibm.com/mx-es/topics/mobile-application-development>
- IBM. (s.f.). Tipos de bases de datos NoSQL [en línea]. Recuperado de <https://www.ibm.com/es-es/topics/nosql-databases>
- IBM. (s.f.). Ventajas de NoSQL [en línea]. Recuperado de <https://www.ibm.com/mx-es/topics/nosql-databases>
- Kielhofner, G. (2004). Terapia ocupacional. Modelo de Ocupación Humana. Teoría y aplicación. Ed. Médica Panamericana. Recuperado de <https://books.google.com.pe/books?id=hBQfhHCs-XIC&printsec=frontcover#v=onepage&q&f=false>



Referencias Bibliográficas

- Loadview (2020). Tipos de pruebas de software: diferencias y ejemplos Recuperado de <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>
- López, S. (2020). Análisis de los requerimientos en habilidades de la industria del software para gerentes de proyectos en entornos ágiles. Desarrollo e Innovación en Ingeniería, 192. Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=8708578>
- Lucidchart. (s.f.). Diagrama de Implementación. Recuperado de <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>
- Medium, Multimedia. (2023). Guía para encontrar programas de estudio en Desarrollo Web en España. Recuperado de <https://www.mediummultimedia.com/web/donde-estudiar-desarrollo-web-en-espana/>
- Medium. (s.f.). ¿Microservicios?. Recuperado de <https://medium.com/@llevit/microservicios-e8c71a6efe52>
- Medium. (s.f.). Apuntes sobre Arquitecturas Limpias (de Software)— Vol 1. Recuperado de <https://medium.com/@PabloLeonPsi/apuntes-sobre-arquitecturas-limpias-de-software-vol-1-1a91131933e2>
- Molina, D. (2022). Qué es Work in Progress (WIP) en Kanban y cómo utilizarlo. Thinking for Innovation. Recuperado de <https://www.iebschool.com/blog/que-es-work-in-progress-wip-en-kanban-y-como-utilizarlo-agile-scrum/>
- Oracle. (s.f.). ¿Qué es una base de datos? Oracle. [online]. Recuperado de <https://www.oracle.com/database/what-is-database.html>
- Plant web (2003) Mejorando el rendimiento (throughput) con la arquitectura de planta digital PlantWeb. Recuperado de <https://www.emerson.com/documents/automation/training-espanolplantweb-availability-es-41538.pdf>
- Platzi (s.f.). Atributos de calidad de un producto de software Recuperado de <https://platzi.com/tutoriales/1248-pro-arquitectura/5498-atributos-de-calidad-de-un-producto-de-software/>
- PowerData (2016) Métricas de calidad de software: una solución excelente. Recuperado de <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/metricas-de-calidad-de-software-una-solucion-excelente>
- Sentries. (2022). Cycle Time: qué es, cómo medirlo y mejorarlo en desarrollo de software. Sentries. <https://sentries.io/blog/cycle-time-que-es/>



Referencias Bibliográficas

- Servnet (2021). Backend y Frontend, ¿Qué es y cómo funcionan en la programación? Recuperado de https://www.researchgate.net/publication/320914166_Modelo_de_procesos_para_el_desarrollo_del_front-end_de_aplicaciones_web
- Siles, R., & Mondelo, E. (2018). Herramientas y técnicas para la gestión de proyectos de desarrollo PM4R. Banco interamericano de desarrollo, 10. Recuperado de https://indesvirtual.iadb.org/file.php/1/PM4R/Guia%20de%20Aprendizaje%20PMA%20SPA.pdf?fbclid=IwAR0_17MRzWGU-xgLtAlHregQQYcDu4V8vVnAga7GbhpR2dJ0QbezaNZ-ig
- Siu, J. L. R. (2020). Las habilidades blandas como base del buen desempeño del docente universitario. Innova research journal, 5(2), 186-199. Recuperado de <https://dialnet.unirioja.es/servlet/articulo?codigo=7475508>
- Solano, A. C., & Minervino, R. (2007). Motivación para liderar, inteligencia práctica y efectividad de los líderes. Revista Iberoamericana de Diagnóstico y Evaluación-e Avaliação Psicológica, 1(23), 175-199. Recuperado de <https://www.redalyc.org/pdf/4596/459645446010.pdf>
- StackOverflow. (2023). Encuesta sobre tecnologías y frameworks web 2023 [online] . Recuperado de <https://survey.stackoverflow.co/2023/#technology>
- Team Insights (2022). Las 5 mejores métricas para medir el compromiso de los empleados Recuperado de <https://teaminsights.io/es/blog/noticias/las-5-mejores-metricas-para-medir-el-compromiso-de-los-empleados/>
- TechTarget. (s.f.). SOAP [online]. Recuperado de <https://searchmicroservices.techtarget.com/definition/SOAP>
- Tekla (2022). ¿Qué es el desarrollo web?. Recuperado de <https://tekla.io/blog/que-es-desarrollo-web/>
- Valdivia-Caballero, J. J. (2016). Modelo de procesos para el desarrollo del front-end de aplicaciones web. Interfases, (009), 187-208. Recuperado de <https://www.researchgate.net/publication/320914166>
- W3C. (s.f.). Servicio web, [online]. Recuperado de <https://www.w3.org/2002/ws/>
- Wam Glowth Agents (2023). Frameworks en el desarrollo web: las mejores prácticas para tu negocio online Recuperado de <https://www.wearemarketing.com/es/blog/frameworks-en-el-desarrollo-web-las-mejores-practicas-para-tu-negocio-online.html>
- Yang, Y., Saladrigas Medina, H., & Torres Ponjuán, D. (2016). El proceso de la comunicación en la gestión del conocimiento. Un análisis teórico de su comportamiento a partir de dos modelos típicos. Revista universidad y sociedad, 8(2), 165-173. Recuperado de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202016000200022
- Zendesk (2023). Gestión de proyectos: 5 pilares para tener éxito en 2023. Recuperado de <https://www.zendesk.com.mx/blog/gestion-proyectos-que-es/>





¡Síguenos, ponte en contacto!



www.certiprof.com

CERTIPROF® is a registered trademark of Certiprof,
LLC in the United States and/or other countries.