Sobre mi:   
Un hombre con lentes y camisa blanca

El contenido generado por IA puede ser incorrecto.

Soy un joven de 18 años en formación en el **SENA**, donde curso el tecnólogo en **Análisis y Desarrollo de Software**. Me desempeño como **desarrollador con énfasis en el backend**, con sólidos conocimientos en **Java y Spring Boot**, así como en la gestión y diseño de **bases de datos relacionales** para garantizar soluciones eficientes y bien estructuradas.

Además, tengo nociones en el desarrollo frontend utilizando tecnologías como **HTML, CSS y JavaScript**, lo que me permite comprender el ciclo completo de las aplicaciones y colaborar de manera efectiva en equipos multidisciplinarios. Me caracterizo por mi compromiso con la calidad del código, la mejora continua y la búsqueda de soluciones tecnológicas innovadoras que respondan a las necesidades del mercado.

Resúmenes de Artículos

# Artículo 1: Tecnologías Front-end y Back-end en Tendencia.

El texto examina la evolución del desarrollo de software y las tendencias que actualmente definen la industria. Se enfoca en la importancia de lenguajes como JavaScript, TypeScript y Golang en el front-end, mientras que en el back-end resaltan Python, PHP, Ruby y el propio JavaScript, todos ampliamente utilizados por grandes empresas como Google o Meta. No se limita únicamente a los lenguajes, sino que también incluye frameworks y herramientas que optimizan el trabajo y permiten que los proyectos sean más ágiles y de mayor calidad.  
  
Algo que destaca es que no es suficiente con elegir el lenguaje “más popular”, sino que lo crucial es que hay que considerar la eficiencia, la escalabilidad y la experiencia del usuario. Igualmente, se menciona lo crucial que es la colaboración entre front-end y back-end para conseguir soluciones integrales y duraderas. El articulo subraya que además de la competencia técnica, lo que realmente importa es la habilidad para adaptarse a nuevas tendencias y utilizar el software libre como un recurso valioso.

# Grafica



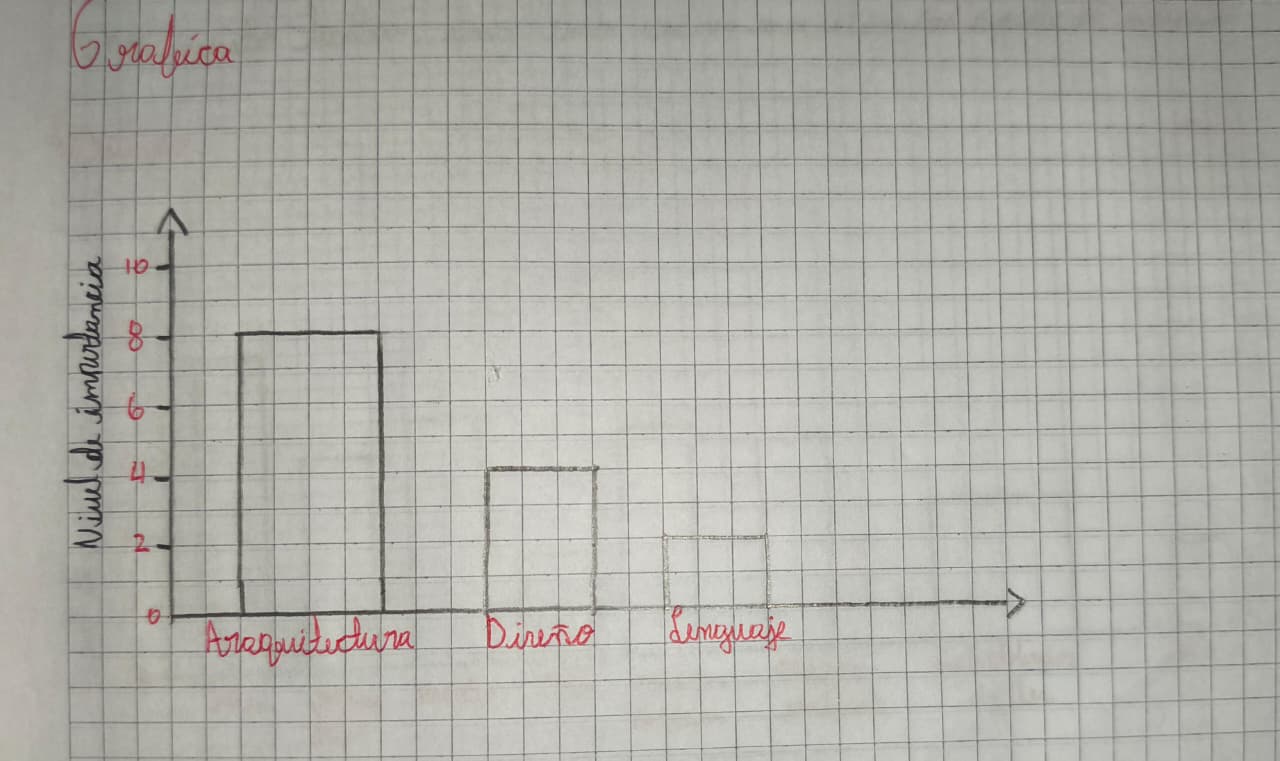
# Reflexión

Este artículo me recordó que lo importante no es solo aprender un lenguaje, sino comprender su propósito y cómo se conecta con otros. Adaptarse y seguir aprendiendo es lo que nos mantiene relevantes en el mundo digital.

Referencia:  
https://repository.usta.edu.co/items/d5041292-ae8f-4b01-a066-849fcb70e15f

# Artículo 2: Especificando una arquitectura de software

El artículo nos habla de la relevancia de los patrones arquitectónicos en el diseño de software, destacando cómo aportan soluciones prácticas y probadas a problemas comunes en proyectos complejos. Explica que estos patrones no solo ayudan a organizar mejor el código, sino que también favorecen la seguridad, la escalabilidad y la rapidez en la entrega de resultados. A lo largo del texto se presentan diferentes tipos de patrones, como los de arquitectura, diseño y lenguaje, además de clasificaciones ampliamente reconocidas como POSA y PEAA.  
  
También se describen diversas formas en que los patrones pueden combinarse, ya sea de manera independiente, como complementos o en secuencia, para adaptarse a las necesidades específicas de cada proyecto. El articulo hace énfasis en la importancia de contar con documentación clara y bien estructurada, ya que esta facilita tanto la implementación como el mantenimiento a largo plazo.

Grafico:  
  


Reflexión:  
Este capítulo me hizo ver que la arquitectura de software es como el plano de una ciudad: define cómo se conectan las partes y asegura que todo funcione de forma ordenada. Los patrones son herramientas que nos ayudan a tomar decisiones más claras y a construir sistemas sólidos desde la base.

Referencia:  
https://revistas.udistrital.edu.co/index.php/tia/article/view/18076

# Artículo 3: Practicante en lenguaje de programación JAVA y nuevas tecnologías

Este artículo analiza cómo estudiantes y profesionales perciben el uso de frameworks en el desarrollo backend, mostrando las diferencias entre la visión académica y la experiencia en el campo laboral. A partir de encuestas y entrevistas, se encontró que los estudiantes valoran más la facilidad de aprendizaje, la versatilidad y el apoyo de la comunidad, mientras que los expertos priorizan aspectos como la seguridad, la eficiencia y la escalabilidad, fundamentales en proyectos empresariales.  
  
El articulo revela una clara brecha entre lo que se enseña en la academia y lo que demanda la industria, lo que resalta la necesidad de generar una mayor integración entre ambos entornos lo que nos proponen es incorporar experiencias reales en los procesos de formación, promover la investigación aplicada y fortalecer la colaboración entre universidades y empresas, de este modo, los futuros desarrolladores no solo adquirirían conocimientos técnicos, sino también una visión más clara con respecto al mercado.

Grafico:  
Imagen que contiene texto, pizarrón

El contenido generado por IA puede ser incorrecto.

reflexión:  
Este capítulo demuestra que una práctica profesional bien planificada permite aplicar conocimientos técnicos en un entorno real. La combinación de tecnologías modernas y metodologías ágiles fortalece el aprendizaje y prepara para afrontar retos del mundo laboral con mayor confianza.

Referencias:  
https://repositorio.utp.edu.co/entities/publication/192d53ab-1ac8-4829-ab8f-2639b995d120

# Artículo 4: Análisis prospectivo de la industria de desarrollo de software en Colombia

El articulo ofrece un análisis prospectivo de la industria del software en Colombia, explorando sus principales retos y oportunidades de crecimiento. Se reconoce que la demanda de soluciones tecnológicas va en aumento y que el país tiene potencial para consolidarse como un actor relevante en el mercado global. Entre los factores positivos se encuentran la adopción de metodologías ágiles, la internacionalización de servicios y la necesidad de talento altamente calificado.  
  
Sin embargo, también se mencionan dificultades importantes como la brecha de habilidades, la competencia con mercados más avanzados y la adaptación a nuevas tecnologías como IA, nube y ciberseguridad. El articulo también nos propone que la clave para el desarrollo del sector está en fortalecer la educación, incentivar la innovación y generar alianzas sólidas entre academia, empresas y Estado.

Grafica:

Imagen que contiene Escala de tiempo

El contenido generado por IA puede ser incorrecto.

Reflexión:  
Este análisis muestra que el futuro de la industria del software en Colombia depende de innovar y actualizarse constantemente, mientras se gestionan retos externos como la ciberseguridad, la formación de talento y la sostenibilidad. La clave está en anticiparse a los cambios y fortalecer alianzas que impulsen el crecimiento.

Referencia:  
https://revistas.poligran.edu.co/index.php/puntodevista/article/view/1415

# Artículo 5: Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software

Este artículo realiza una comparación entre las metodologías, señalando sus ventajas y limitaciones en diferentes contextos. Se explican enfoques clásicos como el modelo en cascada, caracterizando su rigidez y planificación secuencial, frente a metodologías ágiles como Scrum y Kanban, que se centran en la flexibilidad, la retroalimentación continua y la colaboración del equipo. El articulo nos dice que, aunque las metodologías tradicionales ofrecen mayor control y documentación, resultan menos efectivas en entornos cambiantes o con necesidades de entregas rápidas. En conclusión, las ágiles permiten responder mejor a la incertidumbre y facilitan la adaptación durante el ciclo de desarrollo. Sin embargo, se aclara que ninguna metodología es universalmente superior, ya que la elección depende del tipo de proyecto, el equipo de trabajo y los objetivos estratégicos.

Grafica:

Diagrama

El contenido generado por IA puede ser incorrecto.

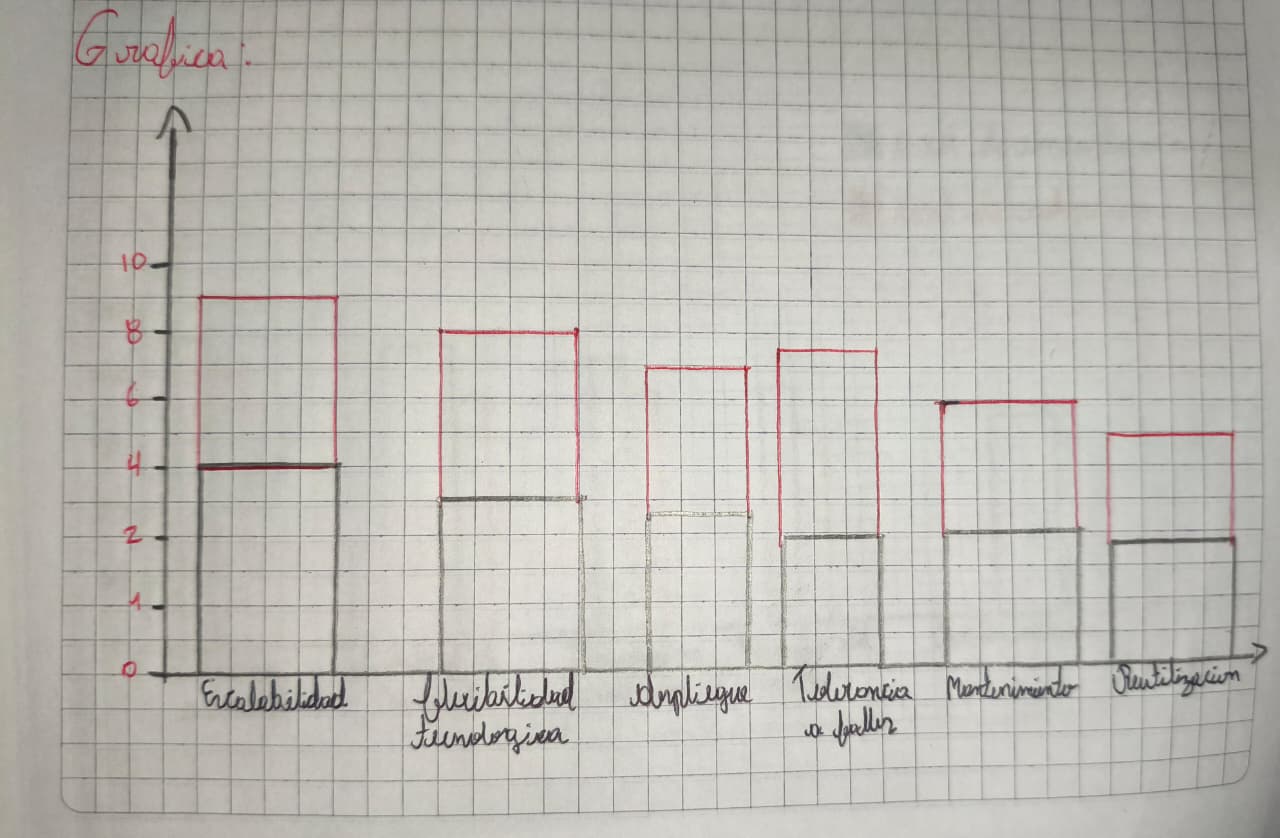
Reflexión:  
Este artículo deja claro que no existe una metodología única para todos los proyectos. Las tradicionales ofrecen control y estructura, mientras que las modernas priorizan adaptabilidad y entregas rápidas. La clave está en elegir la que mejor se ajuste a las necesidades y al contexto del desarrollo.

Referencia:  
https://revistas.pascualbravo.edu.co/index.php/cintex/article/view/334

# Artículo 6 - Estudio sobre metodologías de desarrollo y su impacto en la productividad

Este artículo analiza cómo diferentes metodologías de desarrollo influyen en la productividad de los equipos de software. Explica que los métodos tradicionales, como el modelo en cascada, permiten un mayor control en proyectos bien definidos, pero presentan rigidez frente a cambios. En contraste, las metodologías ágiles favorecen la adaptabilidad, la entrega continua y la retroalimentación frecuente.  
  
El articulo señala que la elección de la metodología depende tanto del tipo de proyecto como de la cultura del equipo, destacando que no existe un único enfoque universal. También se enfatiza que la productividad no debe medirse únicamente por la rapidez, sino también por el valor entregado al usuario final. En conclusión, el artículo plantea que la clave está en equilibrar estructura y flexibilidad para alcanzar mejores resultados en el desarrollo de software.

Grafica:

  
  
reflexión:

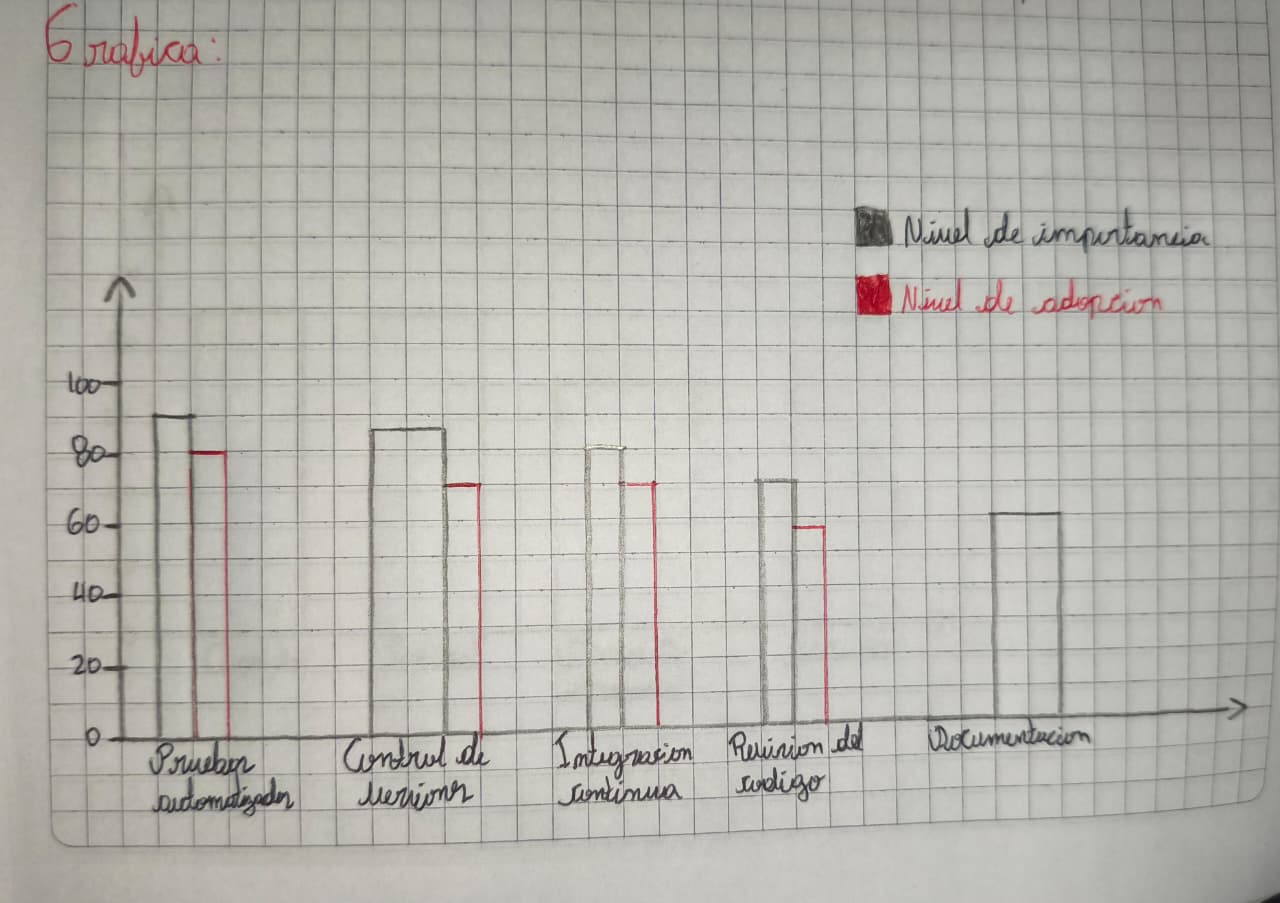
Este capítulo muestra que la industria de software en Colombia tiene un potencial de crecimiento importante, impulsado por la demanda de soluciones en la nube, aplicaciones móviles y servicios de ciberseguridad. La clave estará en innovar y adaptarse a las tendencias globales para mantener la competitividad.

Referencia:   
https://revistas.udistrital.edu.co/index.php/tia/article/view/13364

# Artículo 7 - Buenas prácticas en la construcción de software

Este artículo reflexiona sobre el papel del software libre y de código abierto en la industria tecnológica. Explica cómo estas alternativas se han convertido en una opción viable frente a soluciones propietarias, gracias a su flexibilidad, menor costo y la posibilidad de personalizar las herramientas según las necesidades de cada organización. También se menciona que el software libre fomenta la innovación y el aprendizaje colaborativo, pues permite a los desarrolladores revisar, mejorar y compartir código.  
  
El articulo nos resalta que muchas empresas han adoptado este tipo de soluciones como parte de sus estrategias tecnológicas, reconociendo que no solo representan un ahorro, sino que también favorecen la independencia frente a proveedores. No obstante, se advierte que el uso de software libre implica desafíos, como la necesidad de personal capacitado y el compromiso de mantener la seguridad de las aplicaciones.

Grafica:

  
  
Reflexión:  
Este capítulo resalta que las buenas prácticas no solo mejoran la calidad del software, sino que también facilitan el trabajo en equipo y reducen errores. Adoptarlas de forma constante es clave para construir sistemas más confiables y sostenibles.

Referencia:  
https://revistas.udistrital.edu.co/index.php/tia/article/view/21794

# Artículo 8 - Modelos de procesos de desarrollo aplicado a un proyecto de arquitectura de software

El artículo aborda el tema de la ética en el desarrollo de software, planteando cómo los profesionales de esta área tienen una responsabilidad que va más allá de lo técnico. Explica que las decisiones en torno al diseño y construcción de aplicaciones impactan directamente en la sociedad, lo que hace necesario actuar con principios éticos claros. Se discuten aspectos como la privacidad de los datos, la seguridad de la información y la transparencia en el uso de las tecnologías.  
  
El articulo también resalta la importancia de que los desarrolladores sean conscientes del poder que tienen sus creaciones y del impacto social que pueden generar, tanto positivo como negativo. También, señala que la ética no debe verse como una limitación, sino como una guía para construir software más confiable y responsable.

Grafica:

Esquemático

El contenido generado por IA puede ser incorrecto.  
  
Reflexion:  
Este capítulo me dejó claro que no hay un “modelo perfecto”: depende del contexto. Para definir arquitectura con requisitos cambiantes y alta necesidad de validación temprana, los prototipos ganan por su flexibilidad; la espiral suma cuando el riesgo pesa mucho. Lo importante es elegir con intención, no por costumbre.

Referencia:  
https://revistas.udistrital.edu.co/index.php/tia/article/view/22231

# Artículo 9 - Ética en el desarrollo de software

El artículo se centra en la ética dentro del desarrollo de software, resaltando que los profesionales de esta área tienen una responsabilidad social que va más allá de lo técnico. Se discuten temas como la privacidad de los datos, la transparencia en los algoritmos y el impacto social de las aplicaciones creadas. También se menciona que la ética debe guiar la toma de decisiones para garantizar que la tecnología no se utilice con fines perjudiciales.  
  
El articulo nos habla de que los desarrolladores deben ser conscientes del alcance de sus creaciones, ya que estas pueden generar tanto beneficios como riesgos para la sociedad. La ética no se presenta como una limitación, sino como un marco necesario para orientar la innovación de forma responsable. En conclusión, el artículo plantea que un software verdaderamente exitoso no solo se mide por su funcionalidad, sino también por su compromiso con el bienestar social y la equidad.

Grafica:

Imagen que contiene texto, pizarrón

El contenido generado por IA puede ser incorrecto.  
  
Reflexion:  
  
Este capítulo muestra que un cuadro de mando integral bien diseñado no solo organiza la información, sino que la convierte en conocimiento útil para la toma de decisiones. La clave es crear sistemas éticamente correctos para evitar lios legales y demás cosas.

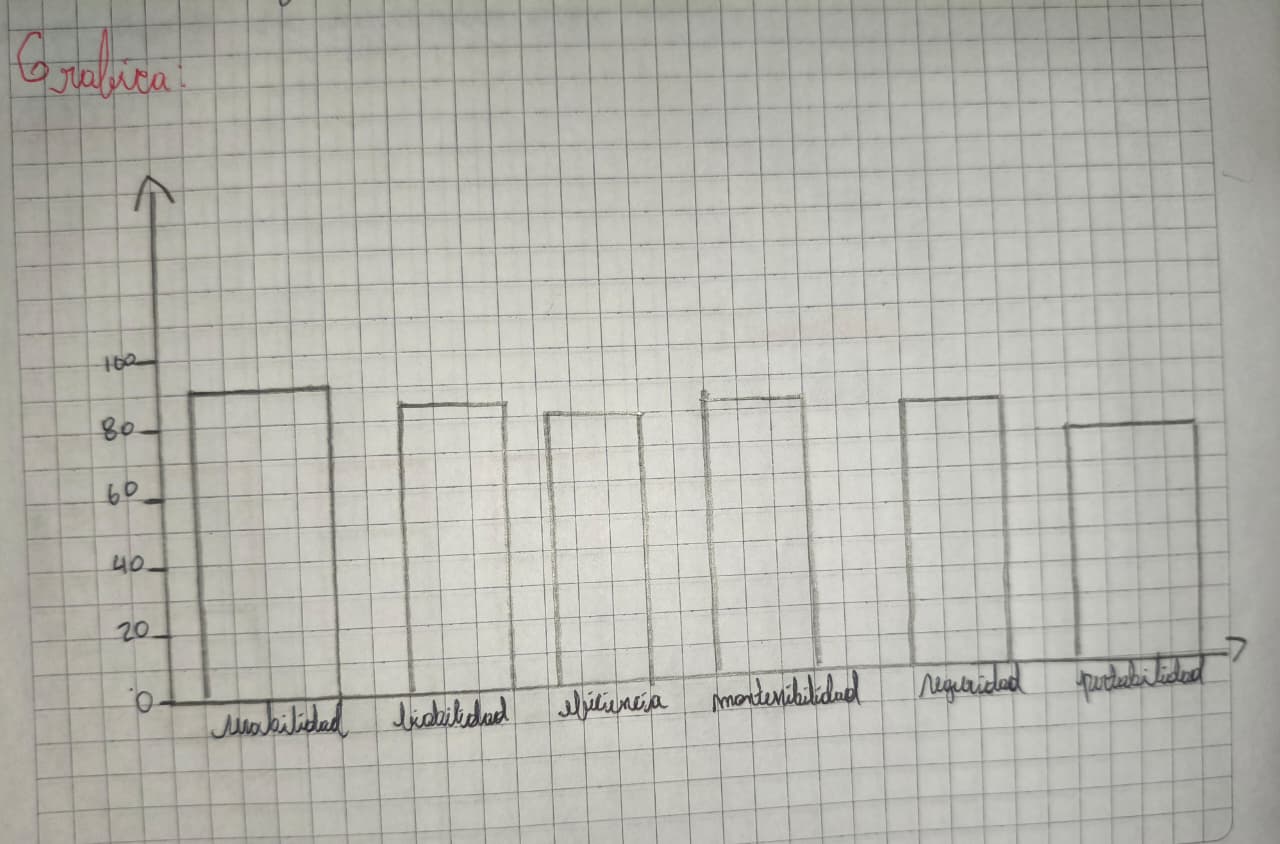
Referencia:  
https://revistas.udistrital.edu.co/index.php/tia/article/view/8766

# Artículo 10 - Estándares para la calidad de software

Este artículo se centra en la calidad del software y los estándares que permiten garantizarla. Explica que, en un mercado cada vez más exigente, los usuarios demandan productos confiables que cumplan con sus necesidades, y por eso han surgido modelos de evaluación como las normas ISO/IEC 9126, 14598 y 25000, así como CMMI y SPICE. Estos estándares sirven como guías para medir aspectos como funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

El artículo se enfoca en que los estándares no solo ayudan a auditar software ya implementado, sino también a orientar proyectos en desarrollo, asegurando que los entregables cumplan con expectativas de calidad. Además, se destacan beneficios como mayor eficiencia en los procesos, productos más confiables y la posibilidad de certificar la calidad frente a competidores.

Grafica:

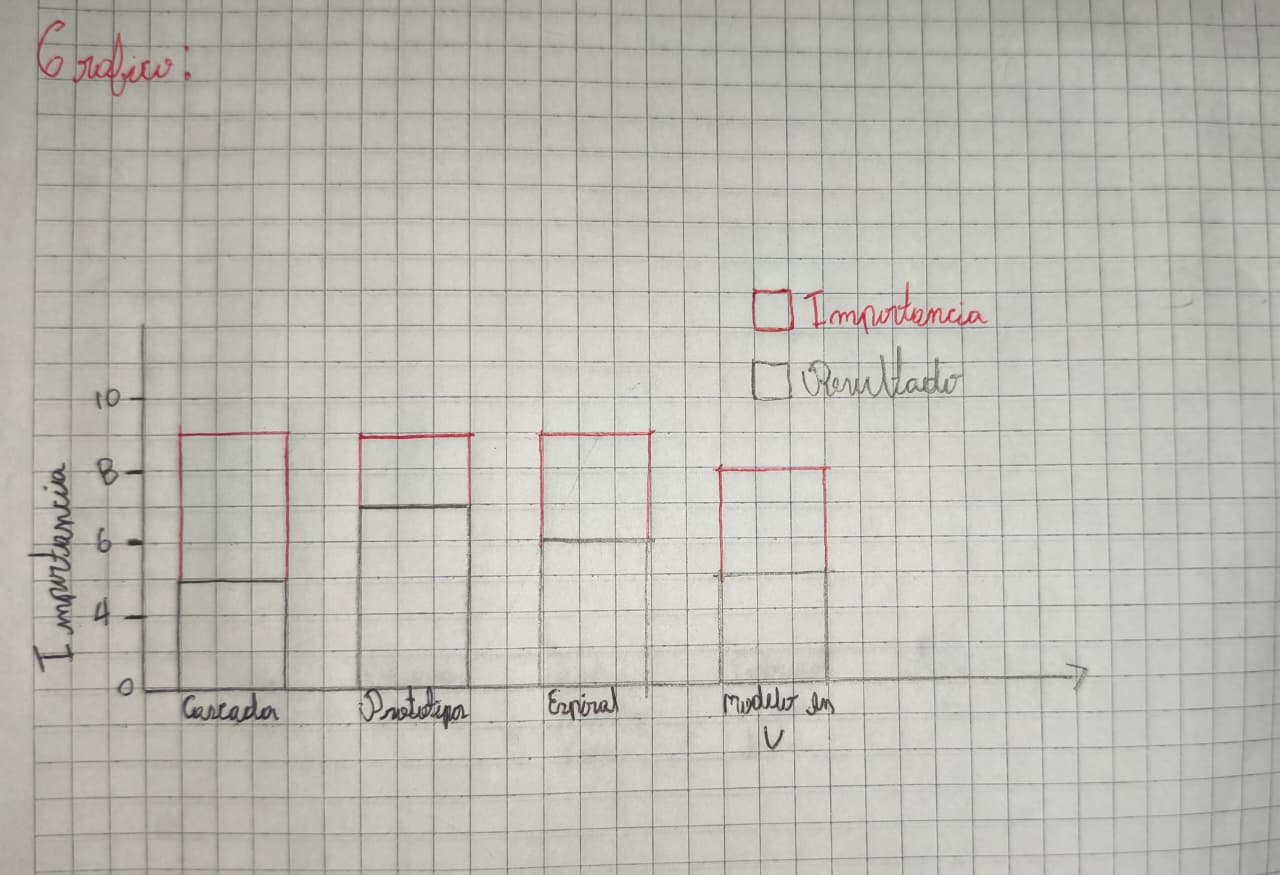


Reflexion:  
Este capítulo resalta que los estándares de calidad de software son la base para desarrollar productos confiables y eficientes. Aplicarlos de forma consistente asegura que el software cumpla con las expectativas y necesidades de los usuarios.

Referencia:  
https://revistas.udistrital.edu.co/index.php/Tecnura/article/view/8066

# Artículo 11 - Modelo de desarrollo de software aplicado al diseño de un prototipo de aplicación web para la gestión de correspondencia en conjuntos residenciales

# El artículo propone un modelo de desarrollo de software aplicado al diseño de un prototipo web para administrar la correspondencia en conjuntos residenciales. Parte de la necesidad de mejorar la organización y trazabilidad de los documentos que ingresan, pues en muchos casos se pierden o no llegan a tiempo a los destinatarios. El prototipo incluye módulos para registrar, clasificar, consultar y generar reportes sobre la correspondencia recibida. Se diseñaron perfiles de usuario como administrador, portero y residente, cada uno con funciones específicas. En pruebas iniciales, se evidenció que la herramienta reduce tiempos de entrega y evita pérdidas, mostrando su utilidad como apoyo en la gestión comunitaria.

Grafico:  


# Reflexión: Este capítulo evidencia que, para un prototipo de aplicación web de gestión de correspondencia, los modelos iterativos como Prototipos y Espiral ofrecen mayor flexibilidad y mejores resultados, mientras que modelos más rígidos como Cascada pueden limitar la adaptación a cambios.

Referencia:  
https://revistas.udistrital.edu.co/index.php/tia/article/view/8388

# Artículo 12 - Scrum a la colombiana

Este artículo analiza cómo se aplica Scrum en equipos colombianos, destacando adaptaciones realizadas frente al modelo original. En muchos casos, los roles de Product Owner y Scrum Master se combinan, los sprints se flexibilizan según la disponibilidad del cliente y las ceremonias suelen acortarse. Aunque estas modificaciones permiten entregas rápidas y mejor comunicación, también generan riesgos como pérdida de formalidad o acumulación de errores. Se concluye que la clave está en mantener la esencia de Scrum, con una definición clara de responsabilidades y criterios de calidad, para evitar caer en prácticas híbridas poco efectivas.

Grafico:  
Imagen que contiene Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Reflexión:  
Este capítulo evidencia que Scrum en Colombia mantiene la esencia del marco original, pero se adapta a la cultura y a las dinámicas de las empresas. Aunque la importancia de las prácticas es alta, la adopción plena aún es un reto, lo que abre oportunidades para mejorar la disciplina y la consistencia en su aplicación.

Referencia:  
https://revistas.udistrital.edu.co/index.php/tia/article/view/9692

# Artículo 13 - Innovación y edad de las empresas: un estudio en empresas argentinas de software

El articulo estudia la relación entre la edad de las empresas de software argentinas y su capacidad para innovar. Los resultados muestran que las empresas jóvenes son más propensas a arriesgarse con nuevas tecnologías y formas de trabajo, mientras que las más antiguas ofrecen mayor estabilidad y experiencia, aunque suelen ser más conservadoras. El artículo señala que la innovación no depende solo de la edad, sino de factores como la cultura organizacional, la disponibilidad de recursos y la apertura al cambio. En conclusión, una combinación de experiencia y flexibilidad es lo que realmente favorece la innovación en el sector.

Grafica:

Gráfico de barras

El contenido generado por IA puede ser incorrecto.  
  
  
Reflexion:  
Este capítulo evidencia que las empresas más jóvenes muestran mayores niveles de innovación y una mayor proporción de inversión en I+D. A medida que aumenta la antigüedad, la innovación tiende a disminuir, lo que sugiere la importancia de mantener una cultura innovadora a lo largo del tiempo para seguir siendo competitivos.

Referencia:  
https://revistas.pucp.edu.pe/index.php/360gestion/article/view/23056

# Artículo 14 - Software educativo para la formulación de proyectos de investigación

Este artículo describe el desarrollo de un software educativo pensado para guiar a estudiantes en la elaboración de proyectos de investigación. La herramienta organiza el proceso en módulos que abarcan desde el planteamiento del problema hasta la metodología, objetivos y cronograma. También ofrece plantillas, ejemplos y listas de verificación para mejorar la coherencia del trabajo. En las pruebas con estudiantes, se observó que el software ayudó a estructurar ideas de trabajo de manera clara y ordenada, aunque no sustituye la orientación del docente. En conclusión, se presenta como un recurso de apoyo que facilita el aprendizaje y la práctica en la formulación de proyectos.

Grafica:  
Imagen que contiene Escala de tiempo

El contenido generado por IA puede ser incorrecto.

Reflexion:  
Este capítulo evidencia que un software educativo para formular proyectos de investigación debe priorizar una interfaz amigable, facilidad de uso y una estructura guiada que oriente al usuario. La retroalimentación oportuna complementa el proceso, mejorando la calidad y efectividad de los proyectos.

Referencia:  
https://revistas.pucp.edu.pe/index.php/educacion/article/view/27455

# Artículo 15 - Calidad en la especificación de requerimientos de software aplicado en metodologías ágiles

El articulo nos habla de la importancia de definir con claridad los requerimientos en proyectos que aplican metodologías ágiles. Se destaca el papel de las historias de usuario, los criterios de aceptación y la validación constante con el cliente. Aunque la documentación es mínima, se enfatiza que debe ser clara y verificable para evitar retrabajos. El artículo propone buenas prácticas como mantener trazabilidad ligera, validar historias en cada sprint y fomentar la comunicación con los usuarios. Concluye que la calidad de los requerimientos en ágil se basa en la precisión y la colaboración, más que en extensos documentos.

Grafico:  
Imagen que contiene Escala de tiempo

El contenido generado por IA puede ser incorrecto.

Reflexion:  
Este capítulo resalta que, en metodologías ágiles, la calidad de los requerimientos es clave para el éxito del proyecto. Mantener claridad, consistencia, viabilidad y completitud asegura que el equipo trabaje alineado y que el producto final cumpla con las expectativas del cliente.

Referencia:  
https://ojs.umsa.bo/ojs/index.php/inf\_fcpn\_pgi/article/view/49

# Artículo 16 - Estudio de calidad y eficiencia de un enfoque de desarrollo software secuencial con programadores solos y en pareja

Este estudio compara el desempeño de programadores trabajando solos frente a aquellos que lo hacen en parejas dentro de un enfoque secuencial. Se evaluaron factores como tiempo de desarrollo, calidad del código y detección de errores. Los resultados muestran que el trabajo en pareja permite identificar fallos más rápido y producir software más robusto, aunque al inicio puede requerir más tiempo. En contraste, los desarrolladores individuales avanzan rápido, pero tienden a dejar errores sin detectar. El artículo concluye que el trabajo en pareja es más eficiente en términos de calidad, aunque exige mayor coordinación.

Grafico:  
Diagrama

El contenido generado por IA puede ser incorrecto.

Reflexion:  
Este capítulo muestra que la calidad en el software educativo secuencial (tipo cascada) no degrada la calidad frente a un proceso ad-hoc, y puede mejorar la eficiencia al reducir compilaciones/ejecuciones necesarias. Las diferencias entre solos y en pareja no son significativas en calidad.

Referencia:  
https://www.scielo.cl/scielo.php?pid=S0718-33052019000200304&script=sci\_arttext

# Artículo 17 - Software Educativo como Método Didáctico en la Enseñanza de las Tablas de Multiplicar

El artículo presenta un software educativo diseñado para mejorar la enseñanza de las tablas de multiplicar en niños de primaria. Basado en un enfoque lúdico, incluye actividades dinámicas, retroalimentación inmediata y recursos visuales y auditivos. La validación con estudiantes mostró que el software incrementa la motivación, facilita la comprensión y mejora el desempeño en evaluaciones posteriores. Se concluye que el uso de herramientas interactivas en el aula puede potenciar el aprendizaje, siempre que sean acompañadas por la guía del docente.

Grafico:  
Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Reflexion:  
Este capítulo muestra que el uso de software educativo para enseñar las tablas de multiplicar potencia el aprendizaje al combinar interactividad, práctica constante y motivación. La comprensión mejora cuando la herramienta es atractiva y fácil de usar, reforzando el aprendizaje de forma lúdica y efectiva.

Referencia:  
https://ojs.docentes20.com/index.php/revista-docentes20/article/view/242

# Artículo 18 - Desarrollo de un software web para la generación de planes de gestión de riesgos de software

Este artículo propone una aplicación web que facilita la elaboración de planes de gestión de riesgos en proyectos de software. La herramienta permite registrar riesgos, calcular probabilidad e impacto, y generar estrategias de mitigación. También ofrece reportes de seguimiento que ayudan a la toma de decisiones. En pruebas aplicadas a proyectos reales, se evidenció que el sistema estandariza el proceso, mejora la trazabilidad y agiliza la documentación. Se concluye que este tipo de herramientas es útil especialmente en equipos con poca experiencia en gestión de riesgos.

Grafica:  
Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.  
Reflexion:

Este capítulo muestra que un software web para la gestión de riesgos debe priorizar una interfaz clara, opciones de personalización y una identificación precisa de riesgos. La documentación completa y actualizada complementa el proceso, asegurando que los planes de mitigación sean efectivos y fáciles de implementar.

Referencia:  
https://www.scielo.cl/scielo.php?pid=S0718-07642020000300135&script=sci\_arttext

# Artículo 19 - Metodologías Ágiles de Desarrollo un Caso de Aplicación Medellín – Colombia 2016 – 2017

Este caso de estudio describe la implementación de metodologías ágiles en un equipo de desarrollo de Medellín. Antes de la adopción, la empresa enfrentaba retrasos, problemas de comunicación y baja satisfacción de clientes. Con la aplicación de Scrum, se mejoró la productividad, la transparencia y la alineación con las necesidades del usuario. Aunque se presentaron retos como la deuda técnica y cambios de alcance, las prácticas de revisión continua y demos frecuentes ayudaron a mitigarlos. En conclusión, la experiencia demostró que la disciplina y la adaptación del equipo fueron claves para lograr resultados positivos.

Grafica:  
Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.

Reflexion:  
Este capítulo muestra que la aplicación de metodologías ágiles en Medellín permitió mejorar la comunicación, la calidad del producto, la entrega a tiempo y la productividad. Aunque la importancia de estos factores es alta, el nivel de satisfacción indica que aún hay espacio para optimizar la implementación y consolidar la cultura ágil en las organizaciones.

Referencia:  
https://revistas.ucatolicaluisamigo.edu.co/index.php/lampsakos/article/view/2521

# Artículo 20 - Metodología iterativa de desarrollo de software para microempresas

El artículo propone una metodología iterativa adaptada a las condiciones de las microempresas, caracterizadas por recursos limitados y necesidades cambiantes. Se plantean ciclos cortos de desarrollo, entregas funcionales y retroalimentación constante del cliente. La propuesta busca equilibrar simplicidad y flexibilidad, garantizando que los resultados sean útiles en plazos reducidos. Se concluye que esta metodología es una alternativa viable para microempresas que requieren soluciones tecnológicas rápidas y ajustadas a sus posibilidades.

Grafica:

Diagrama

El contenido generado por IA puede ser incorrecto.

Reflexion:  
Este capítulo evidencia que la metodología iterativa es especialmente útil para microempresas, ya que permite adaptarse rápidamente a cambios, mantener prioridades claras y recibir retroalimentación constante. Esto mejora la eficiencia y asegura que el producto final responda mejor a las necesidades reales del cliente.

Referencia:  
https://www.scielo.sa.cr/scielo.php?pid=S0379-39822015000300099&script=sci\_arttext

# 