**Introducción**

En el mundo del desarrollo de software, gestionar bien los recursos y optimizar los procesos es esencial para que las organizaciones sigan siendo sostenibles y competitivas. Sin embargo, los equipos de desarrollo a menudo se enfrentan a la sobrecarga de trabajo, un problema que no solo afecta la productividad sino que también desperdicia recursos valiosos. Este problema aparece de muchas formas, como la multitarea ineficiente, errores en los requisitos, retrabajos, la implementación de características innecesarias y la falta de comunicación entre los equipos.

El desperdicio de recursos no solo aumenta los costos operativos, sino que también puede desmotivar al equipo y reducir la calidad del producto final. Recientes estudios muestran que hasta el 35% de los errores en el desarrollo de software se deben a requisitos inconsistentes o poco claros, lo que puede significar pérdidas económicas significativas a nivel global. Además, la falta de una adecuada planificación y la implementación apresurada de metodologías como Agile sin la debida preparación pueden agravar estos problemas, perpetuando ciclos de trabajo ineficientes y disminuyendo la satisfacción tanto del cliente como del equipo de desarrollo.

Dada esta situación, es crucial explorar soluciones para enfrentar estos desafíos de manera efectiva. La adopción de metodologías ágiles y Lean, el uso de herramientas de automatización y prácticas de comunicación eficaces son alternativas prometedoras para reducir el desperdicio y optimizar el flujo de trabajo. Estas estrategias no solo permiten una mejor gestión de los recursos, sino que también fomentan la colaboración y se centran en la entrega continua de valor, factores clave para el éxito en un entorno competitivo y en constante cambio.

Este estudio se centrará en analizar las causas del desperdicio de recursos en el desarrollo de software debido a la sobrecarga de trabajo y propondrá medidas concretas para mitigar su impacto. Combinando perspectivas técnicas, organizacionales y humanas, el objetivo es ofrecer una visión integral que permita a las organizaciones no solo reducir costos y mejorar la calidad del software, sino también promover entornos de trabajo más sostenibles y motivadores.

**Resumen**

La investigación aborda el desperdicio de recursos en el desarrollo de software, provocado por sobrecarga de trabajo. Este fenómeno incluye problemas como multitarea, defectos en requisitos, retrabajos y la implementación de funcionalidades innecesarias, que afectan tanto la productividad como la moral del equipo. Se propone un enfoque integral que combina la adopción de metodologías ágiles, el uso de herramientas de automatización y la mejora en la comunicación organizacional. Los hallazgos destacan que prácticas eficientes pueden reducir costos, mejorar tiempos de entrega y elevar la calidad del software, fortaleciendo la sostenibilidad operativa y la competitividad de las organizaciones.

**Palabras clave**

Desarrollo de software, desperdicio de recursos, sobrecarga de trabajo, metodologías ágiles, optimización de procesos, automatización, eficiencia operativa

**CAPÍTULO I**

**EL PROBLEMA**

* 1. **Tema**

“Desperdicio de recursos en el desarrollo de software provocado por sobrecarga de trabajo” 

* 1. **Planteamiento de problema**  
       
     **Contextualización**

El desperdicio en el desarrollo de software puede tener un impacto económico significativo debido a costos innecesarios asociados con errores y retrabajos. Las empresas pueden ver aumentados sus costos operativos y reducir sus márgenes de beneficio si no identifican y eliminan estos desperdicios de manera efectiva. Implementar prácticas de desarrollo eficientes puede llevar a un uso más inteligente de los recursos financieros, reduciendo así los costos totales del proyecto.

La cultura de la organización juega un papel crucial en la cantidad de desperdicios presentes en el proceso de desarrollo. Una cultura que fomenta la comunicación abierta, la colaboración y la mejora continua puede ayudar a identificar y eliminar desperdicios más rápidamente. Por el contrario, una cultura rígida y burocrática puede perpetuar prácticas ineficientes y dificultar la adopción de métodos ágiles y Lean.

 La mala priorización de proyectos puede llevar a un desperdicio de recursos, ya que se asignan esfuerzos y tiempo a tareas que no aportan valor significativo al producto final. La implementación de metodologías como Agile puede ayudar a las organizaciones a priorizar las tareas de manera más efectiva, asegurando que los equipos trabajen en los aspectos más críticos y valiosos del proyecto.

La multitarea y la burocracia pueden ocultar y perpetuar prácticas ineficientes en el desarrollo de software. La tendencia a manejar múltiples proyectos simultáneamente puede dispersar la atención y los recursos, reduciendo la eficiencia y aumentando el riesgo de errores. Reducir la burocracia innecesaria y fomentar enfoques más enfocados puede mejorar significativamente la eficiencia del equipo.

El retrabajo debido a versiones de código rápidamente hechas y luego corregidas puede afectar la productividad de los desarrolladores, ya que obliga a dedicar tiempo a corregir errores que podrían haberse evitado inicialmente con una planificación y revisión más cuidadosa. Este ciclo de corrección perpetua no solo consume recursos, sino que también puede desmotivar al equipo.

Los defectos en el software pueden consumir recursos adicionales para su corrección, especialmente si estos defectos no se detectan hasta etapas tardías del desarrollo. La implementación de pruebas continuas y revisiones regulares del código puede ayudar a identificar y corregir estos defectos más temprano, reduciendo así el costo y el impacto de los errores.

 La implementación de funcionalidades innecesarias puede llevar a un desperdicio de recursos, ya que se invierte tiempo y esfuerzo en características que no aportan valor significativo al usuario final. Adoptar un enfoque basado en el valor y priorizar funcionalidades según su impacto puede ayudar a evitar este tipo de desperdicio.

La falta de comunicación efectiva entre los miembros del equipo puede resultar en errores y retrabajos, ya que los desarrolladores pueden no estar alineados con los objetivos y requisitos del proyecto. Fomentar una comunicación clara y regular puede ayudar a asegurar que todos los miembros del equipo tengan una comprensión común de las tareas y expectativas.

Los requisitos mal definidos pueden causar retrabajos y desperdicio de recursos, ya que los desarrolladores pueden trabajar en funcionalidades incorrectas o incompletas. Involucrar a los stakeholders en la definición y revisión de los requisitos puede mejorar la claridad y precisión, reduciendo el riesgo de malentendidos y errores.

La implementación de prácticas Lean y Agile puede ayudar a reducir desperdicios al enfocarse en la entrega continua de valor y la mejora constante de los procesos. Estas metodologías fomentan la colaboración, la flexibilidad y la adaptación, lo que puede reducir el tiempo y los recursos desperdiciados en tareas no esenciales.

La evaluación de la sostenibilidad del software puede identificar áreas de desperdicio, ayudando a las organizaciones a desarrollar productos más eficientes y duraderos. Esto incluye evaluar el impacto ambiental y económico del software, y buscar formas de minimizar el desperdicio en cada etapa del ciclo de vida del producto.

La optimización de procesos puede reducir el desperdicio en diferentes niveles de la organización al identificar y eliminar ineficiencias. Esto puede incluir la automatización de tareas repetitivas, la mejora de la gestión del flujo de trabajo y la implementación de herramientas de seguimiento del rendimiento.

La identificación y eliminación de desperdicios es crucial para mejorar la eficiencia en el desarrollo de software. Las organizaciones pueden utilizar técnicas como el mapeo del flujo de valor para visualizar y analizar sus procesos, identificando áreas donde se desperdician recursos y desarrollando estrategias para abordar estos problemas.

Evitar la suboptimización es esencial para asegurar que las mejoras realizadas en una parte del proceso no causen problemas en otra. Esto implica optimizar todo el proceso de desarrollo y el producto en su conjunto, en lugar de enfocarse únicamente en áreas individuales, para lograr una mejora coherente y holística

La adopción de prácticas modernas en el desarrollo de software, como DevOps y Agile, puede acelerar la transformación digital de una organización. Esto incluye la implementación de herramientas y metodologías que optimizan el desarrollo, reducen los desperdicios y garantizan una entrega más rápida al mercado.

Eliminar actividades innecesarias en el ciclo de vida del software contribuye directamente a mejorar la eficiencia operativa general de la organización. Esto puede incluir reducir tiempos de espera, eliminar pasos redundantes y optimizar la asignación de recursos. ​[1]​

La automatización de procesos no solo reduce el tiempo dedicado a tareas manuales, sino que también mejora la calidad y consistencia del software entregado, reduciendo desperdicios en cada etapa del desarrollo. ​[2], [3]​

Implementar metodologías ágiles permite que los equipos trabajen en ciclos cortos y repetitivos, reduciendo el desperdicio causado por requisitos mal definidos o cambios tardíos. Estas prácticas fomentan una mejor colaboración y un enfoque en la entrega de valor continuo.​[4]​

Promover la colaboración entre equipos multidisciplinarios reduce los tiempos de espera y la duplicación de esfuerzos. Esto se logra mediante reuniones regulares, herramientas colaborativas y la adopción de enfoques integradores.​[1]​

Implementar ciclos de retroalimentación más cortos entre los equipos de desarrollo, pruebas y operaciones asegura que los problemas se identifiquen y resuelvan rápidamente, minimizando la necesidad de retrabajo.​[5]​

Automatizar las validaciones en etapas iniciales del desarrollo ayuda a detectar errores antes de que se conviertan en problemas costosos, reduciendo significativamente los desperdicios.​[2], [6]​

Adoptar un enfoque basado en el valor al seleccionar las funcionalidades evita el desperdicio de recursos en características que no son esenciales para los usuarios finales.​[1]​

La refactorización continua del código asegura que los problemas técnicos se aborden de manera incremental, evitando la acumulación de deuda técnica y manteniendo un código limpio y mantenible.​[4], [6]​

En la industria del software, los defectos en requisitos mal definidos representan un problema grave. Estudios indican que hasta el 35% de los defectos provienen de requisitos inconsistentes o poco claros, causando costos globales estimados en $720 mil millones anuales debido a reelaboraciones evitables. Estos problemas surgen porque muchas organizaciones no priorizan la calidad en la etapa de definición de requisitos, lo que a menudo lleva a sistemas ineficaces y trabajo duplicado.​[7]​

A nivel global, las empresas han adoptado metodologías ágiles sin suficiente preparación, lo que aumenta el retrabajo y desperdicio de recursos. Aunque Agile puede mejorar los tiempos de entrega, una implementación deficiente resulta en pérdidas significativas, ya que los equipos no logran mantener un balance entre velocidad y calidad ​[8]​

La creación de funcionalidades "extra" que los usuarios finales no solicitan ni utilizan es una forma de sobreproducción. A nivel global, esto sucede debido a una desconexión entre las expectativas del cliente y las decisiones técnicas, consumiendo recursos que podrían haberse dedicado a resolver problemas reales.​[8]​

En el contexto organizacional, los procesos redundantes son una fuente de desperdicio común. Por ejemplo, en proyectos donde múltiples equipos trabajan en soluciones similares sin colaboración, se duplican esfuerzos. Esto no solo genera costos adicionales, sino que complica la integración final del sistema. ​[9]​

Las historias de usuario que no se desarrollan o implementan generan un "stock" de tareas pendientes que ocupan espacio físico o mental. Esta acumulación dificulta que los desarrolladores se concentren en nuevas tareas, ralentizando los proyectos y desperdiciando horas de planeación inicial. ​[8]​

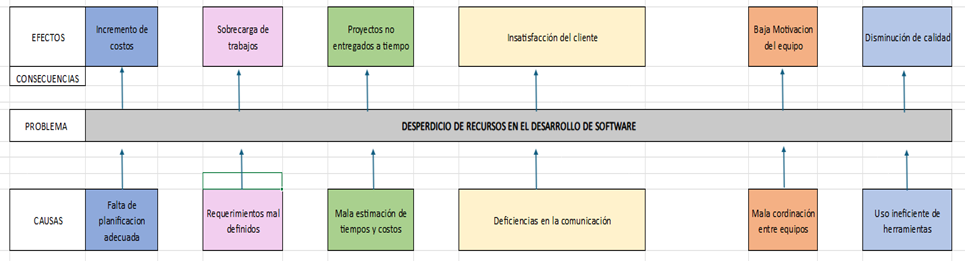
La falta de claridad en la toma de decisiones dentro de los proyectos genera confusión y esfuerzos mal direccionados. Esto puede manifestarse en tareas repetidas o proyectos que cambian de alcance continuamente, causando pérdidas considerables en tiempo y dinero para las organizaciones ​[8], [9]​

A nivel de equipo, los cambios de contexto frecuentes (por ejemplo, de una tarea a otra sin completar la inicial) resultan en pérdida de eficiencia. Cada cambio obliga al desarrollador a recalibrar su enfoque, lo que puede consumir hasta un 20% del tiempo productivo diario.​[10]​

Los desarrolladores suelen enfrentarse a retrasos causados por dependencias, como esperar aprobaciones de superiores o que otros equipos completen su parte del trabajo. Este tiempo de inactividad desperdicia horas que podrían haberse aprovechado mejor con planificación más ágil.​[8]​

Muchos equipos pierden tiempo en tareas repetitivas, como buscar documentos en rutas de directorio complicadas o realizar pruebas manuales que podrían ser automatizadas. Estas acciones no aportan valor directo al producto final y podrían eliminarse mediante herramientas adecuadas.​[8]​

* + 1. **Análisis crítico**

**Árbol de problemas**

* **Falta de planificación adecuada e incremento de costos**

La falta de planificación eficiente en proyectos puede llevar a un uso ineficiente de los recursos, incrementando los costos por sobretiempo, errores o la necesidad de repetir trabajos. La planificación inadecuada también afecta la distribución de personal y recursos, lo que genera ineficiencias operativas y financieras. Un enfoque efectivo incluye anticipar necesidades y alinear los recursos con los objetivos estratégicos. ​[11], [12]​

* **Requerimientos mal definidos y sobrecarga de trabajos**

Definir correctamente los requerimientos es esencial para evitar sobrecarga laboral y malentendidos. La falta de claridad lleva a un aumento en las revisiones y retrabajos, afectando los tiempos y la calidad. Implementar metodologías ágiles y una comunicación eficaz con los stakeholders puede ayudar a reducir estos problemas ​[13]​

* **Mala estimación de tiempos y costos y proyectos no entregados a tiempo**

Una estimación inexacta de tiempos y costos impacta directamente en el cumplimiento de plazos, generando retrasos y costos adicionales. Es crucial aplicar herramientas como el análisis de costos ABC y técnicas de evaluación de riesgos para obtener presupuestos y cronogramas más precisos ​[11], [14]​

* **Deficiencias en la comunicación e insatisfacción del usuario**

La comunicación ineficiente en los equipos genera confusión, errores y falta de alineación con los objetivos del cliente. Esto puede resultar en entregables que no cumplen con las expectativas, afectando la satisfacción del cliente y la reputación de la organización. Las reuniones periódicas y el uso de herramientas colaborativas pueden mejorar este aspecto ​[11], [12]​

* **Mala coordinación entre equipos y baja motivación del equipo**

La falta de coordinación causa conflictos entre equipos y disminuye la moral, afectando la productividad. Una estructura organizacional clara y la implementación de metodologías de gestión de equipos pueden contrarrestar estos problemas, promoviendo un ambiente de trabajo más eficiente ​[12]​

* **Uso ineficiente de herramientas y disminución de la calidad**

El uso incorrecto o inadecuado de herramientas tecnológicas provoca ineficiencias y una caída en la calidad del producto final. La capacitación adecuada en herramientas y la selección de tecnologías que se ajusten a las necesidades del proyecto son clave para garantizar un desempeño óptimo ​[14]​

* + 1. **Prognosis**

1. **¿Qué pasará o qué pasaría si no se atiende el problema y se deja como está la situación?**
2. Si el desperdicio de recursos en el desarrollo de software causado por la sobrecarga de trabajo no se aborda, las empresas continuarán enfrentando pérdidas significativas debido al retrabajo, defectos en el código y la implementación de funcionalidades innecesarias. Esto incrementará los costos operativos y reducirá la eficiencia general de los equipos. La sobrecarga de trabajo desmotivará al personal, generando agotamiento, insatisfacción y rotación frecuente, lo cual afectará la continuidad y calidad de los proyectos.
3. **¿Qué pasará o qué pasaría si no se intenta dar solución al problema?**
4. En ausencia de una solución, los recursos seguirán siendo mal gestionados, llevando a un aumento en los tiempos de entrega y a una disminución en la calidad del software. Los equipos de desarrollo enfrentarán mayor estrés, lo que incrementará la probabilidad de errores y defectos en las etapas finales del desarrollo. Esto afectará la competitividad de las empresas, limitando su capacidad para responder a las demandas del mercado y comprometiendo la satisfacción del cliente, lo cual podría resultar en la pérdida de proyectos clave.
5. **Conocimiento anticipado de un suceso:** Si no se implementan soluciones, el ciclo de trabajo ineficiente continuará perpetuándose. Las metodologías tradicionales o mal aplicadas seguirán fomentando prácticas que no agregan valor, mientras que la falta de enfoque en tareas prioritarias aumentará los recursos desperdiciados. Con el tiempo, los equipos podrían caer en un estado de agotamiento extremo, comprometiendo la capacidad de las organizaciones para completar proyectos de forma exitosa.
6. **Predicción:** La persistencia de prácticas ineficientes resultará en un deterioro constante de la calidad del software desarrollado, acompañado de costos operativos insostenibles. Las empresas podrían perder su posición en el mercado frente a competidores más ágiles y eficientes que adopten metodologías modernas para minimizar el desperdicio de recursos. Esto podría llevar a un aumento en la insatisfacción de los clientes y una disminución de la confianza en los equipos de desarrollo.
7. **Panorama negativo de no dar atención al problema identificado:** En el peor de los casos, el desperdicio de recursos no controlado, causado por la sobrecarga de trabajo, podría llevar al colapso de los equipos de desarrollo, quienes se verían incapaces de cumplir con los plazos o con los estándares de calidad exigidos. Las empresas enfrentarían pérdidas financieras graves, limitaciones para innovar y una disminución notable en su capacidad para competir en mercados tecnológicos altamente demandantes. Esto generaría una pérdida de oportunidades estratégicas, afectando su sostenibilidad a largo plazo.
   * 1. **Formulación del problema**

¿Cómo impacta la sobrecarga de trabajo en el desarrollo de software y qué medidas pueden tomarse para reducir el desperdicio de recursos asociado?

* + 1. **Preguntas directrices**

**Tema:** El desperdicio de recursos por la sobrecarga de trabajo en el desarrollo de software.

**Variable independiente:** Desperdicio de recursos

* + - 1. ¿Qué se entiende por desperdicio de recursos en el desarrollo de software?
      2. ¿Cuáles son las principales causas del desperdicio de recursos en proyectos de software?
      3. ¿Cómo afecta el desperdicio de recursos al rendimiento laboral?
      4. ¿Qué medidas pueden tomar para minimizar el desperdicio de recursos?
      5. ¿Qué tipo de recursos suelen ser desperdiciados con mayor frecuencia?

**Variable dependiente:** Sobrecarga de trabajos

¿Qué es la sobrecarga de trabajo?

¿Por qué se manifiesta la sobrecarga de trabajo?

¿Qué efectos tiene la sobrecarga de trabajo en la eficiencia del desarrollo de software?

¿Qué factores internos y externos contribuyen al aumento de la sobrecarga de trabajo?

¿Qué estrategias se pueden aplicar para reducir la sobrecarga de trabajo?

¿Cómo puede evitarse la sobrecarga de trabajo?

**Solución:**

1. ¿Qué estrategias pueden implementarse para reducir el desperdicio de recursos en el desarrollo de software?
2. ¿Cómo debería planificarse un proyecto para evitar el desperdicio de recursos?
3. ¿Deberían las empresas priorizar la capacitación de sus equipos para reducir el desperdicio de recursos?
4. ¿Cómo pueden las herramientas de gestión de proyectos optimizar el uso de recursos en desarrollo de software?
5. ¿Qué buenas prácticas en comunicación pueden minimizar la duplicación de esfuerzos?
6. ¿Es la adopción de metodologías como Scrum o Kanban una solución efectiva para manejar el desperdicio de recursos?

* + 1. **Delimitación del objeto de investigación**

**Geográfica**

**Espacial**

**Temporal**

* 1. **Justificación**

Este trabajo de investigación es importante porque aborda un problema crítico en la industria del software: el desperdicio de recursos, que repercute directamente en el costo, el tiempo y la calidad de los productos desarrollados. Identificar las causas del desperdicio y proponer soluciones efectivas puede aumentar la competitividad de las empresas y la satisfacción del cliente, al garantizar procesos más eficientes y sostenibles.

El problema investigado es relevante en el contexto actual de alta demanda tecnológica, donde el uso óptimo de los recursos es clave para cumplir con plazos ajustados y responder a un mercado en constante evolución. Resolver esta problemática puede beneficiar no solo a las empresas de software, sino también a los clientes que dependen de estas soluciones para sus operaciones.

 El tema es innovador porque busca combinar aspectos de gestión, planificación y metodologías modernas para abordar una problemática recurrente en el desarrollo de software. A diferencia de otros estudios enfocados exclusivamente en herramientas o tecnologías, esta investigación propone una visión integral que incluye factores humanos, técnicos y organizativos para optimizar el uso de recursos.

El impacto del trabajo radica en la posibilidad de establecer lineamientos que permitan a las organizaciones reducir el desperdicio de recursos, mejorar la productividad de los equipos y garantizar resultados más consistentes. Al hacerlo, este estudio contribuye a la evolución de la industria del software hacia prácticas más eficientes y sostenibles.

Finalmente, el proyecto es factible debido a los recursos disponibles: acceso a literatura especializada, conocimiento previo en gestión y desarrollo de software, y la posibilidad de trabajar con casos de estudio reales en equipos de desarrollo. Las herramientas necesarias, como software de gestión y metodologías ágiles, están ampliamente disponibles y serán utilizadas para realizar observaciones, encuestas y análisis que sustenten las conclusiones del estudio.

* 1. **Objetivos**  
     1. **Objetivo general:**

Mejorar la gestión de proyectos de desarrollo de software mediante la reducción de desperdicios y la optimización de recursos, utilizando metodologías ágiles, Lean y herramientas de automatización para identificar y eliminar las principales causas de ineficiencia derivadas de la sobrecarga de trabajo.

* + 1. **Objetivos específicos**
  1. Realizar un análisis diagnóstico de las causas principales del desperdicio de recursos en procesos de desarrollo de software.
  2. Examinar que tan efectiva puede resultar la aplicación de metodologías ágiles y Lean al momento de reducir la probabilidad de que exista un desperdicio de recursos en un software.
  3. Proponer la adopción de tecnologías de automatización y herramientas colaborativas para optimizar flujos de trabajo y reducir actividades repetitivas.

**CAPITULO II**  
  
**MARCO TEÓRICO**

* 1. **Investigaciones previas:**

asdsasdasda

* 1. **Marco teórico**

**Constelación de ideas organizador gráfico de variables independientes:**

**Constelación de ideas organizador gráfico de variables dependientes:**

1. **CAPÍTULO III**

**METODOLOGÍA**

* 1. **Matriz metodológica**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tema:** | Errores en la navegación generados por la contaminación visual en la interfaz | | | | | |
| **Objetivo general** | Diagnosticar los errores en la navegación generados por la contaminación visual en la interfaz de las plataformas educativas de la universidad en pro de la mejora de la experiencia de usuario dentro del uso cotidiano. | | | | | |
| **Objetivos específicos** | **Pasos actividades** | **Datos e información necesaria** | **Objeto-sujeto** | **Técnica** | **Instrumento** | **Resultado esperado** |
| Identificar errores de navegación que se produce por contaminación visual dentro de la plataforma universitaria utilizando como medio la encuesta a fin de que el usuario defina los errores que tiene y como le gustaría que se manejen las soluciones. | * Realizar una investigación general de que se define como error de navegación, que tipos de errores se pueden dar dentro del uso de una plataforma educativa * Relacionar la contaminación visual con los errores de navegación * Realizar una encuesta al usuario de la plataforma educativa. | * Errores de navegación * Tipos de errores de navegación * Contaminación visual en espacios web * Preguntas y posibles respuestas del cuestionario. | * Errores de navegación * Contaminación visual * Usuarios de la plataforma | * Revisión bibliográfica * Encuesta | * Repositorios * Bibliotecas virtuales * Matriz de conceptualización * Cuestionario | Se espera reconocer con detalle que es un error de navegación, como puede darse un error de navegación y como se interrelacionan estos con la contaminación visual dentro de la interfaz de la plataforma educativa por medio de la aplicación del cuestionario, dicho cuestionario permitirá que los usuarios den su punto de vista sobre cómo debería organizarse la plataforma para reducir los errores. |
| Definir herramientas para evitar errores de navegación a través de los medios de consulta disponibles y de la experiencia de programadores especialistas en experiencia de usuario. | * Analizar los medios de consulta existentes (documentación, guías de buenas prácticas, tutoriales). * Identificar y documentar las herramientas más recomendadas. * Evaluar la aplicabilidad de estas herramientas en la plataforma universitaria | * Técnicas de diseño limpio * documentación de frameworks * Api de uso dentro de interfaces. * Librerías enfocadas en desarrollo visual. | * UX y UI * Frameworks y librerías de desarrollo front end | * Revisión bibliográfica | * Repositorios * Bibliotecas virtuales * Matriz de conceptualización | Se espera que mediante el análisis de las técnicas de diseño limpio y la documentación de los frameworks disponibles gracias a una correcta revisión bibliográficas se pueda reconocer que herramientas pueden ser usadas dentro de la interfaz de la plataforma educativa de la universidad. |
| Crear un manual de gestión de errores de navegación más comunes generados por la contaminación visual basándose en la información recogida dentro de la investigación mediante la estandarización de las interfaces. | * Analizar los antecedentes indagados (artículos científicos). * Extraer las soluciones más relevantes a errores más comunes hacia los usuarios. * Efectuar un catálogo con todos los resultados obtenidos. | * Antecedentes indagados y conceptos anteriores * Soluciones potenciales * Frameworks * Teoría del color | Errores de navegación  Usuarios  Desarrolladores interesados | * Encuestas * Análisis estadístico | * Cuestionario * Excel | Se espera crear un manual de gestión de errores de navegación que sean comunes dentro de la plataforma y que estén conectados directamente con la contaminación visual tomando en cuenta la experiencia del usuario que fue consultada en objetivos anteriores. |

* 1. **Nivel o tipo de investigación**

La investigación se clasifica como **nivel explicativo** debido a que su propósito es identificar, comprender y detallar las relaciones causales entre la contaminación visual en plataformas educativas y los problemas de navegación más comunes que enfrentan los usuarios. Este nivel de profundidad se refleja en los siguientes aspectos clave:

**Relación causal entre variables**

El enfoque principal radica en explicar cómo y por qué la sobrecarga de estímulos visuales afecta aspectos específicos del comportamiento del usuario, como los tiempos de respuesta, la fatiga cognitiva, el procesamiento de información y la memoria de trabajo. Esto trasciende una mera descripción de los problemas, centrándose en los mecanismos subyacentes y los efectos directos e indirectos de la contaminación visual en la experiencia de los estudiantes.

**Impacto en la mejora y estandarización de interfaces**

Al tener como objetivo principal la creación de un manual de gestión de errores de navegación, la investigación no solo busca describir problemas, sino también ofrecer soluciones prácticas basadas en los hallazgos. Este manual estará fundamentado en la estandarización de las interfaces y ofrecerá directrices claras para evitar la contaminación visual. Dicho enfoque conecta los hallazgos teóricos con aplicaciones prácticas, abordando el "por qué" de los errores y el "cómo" pueden resolverse.

**Ergonomía cognitiva y macro ergonomía**

El estudio considera cómo factores macro ergonómicos y cognitivos, como la interfaz poco intuitiva y la sobreabundancia de opciones, afectan la experiencia y motivación de los estudiantes. La investigación busca explicar cómo la contaminación visual reduce la retención y atención, afectando la interacción humano-máquina de manera integral. Este análisis profundo de las dinámicas cognitivas es típico de una investigación explicativa.

**Contexto educativo y mejora de la experiencia del usuario**

En el marco de las plataformas educativas, definidas como un "conjunto de soluciones de software" que facilitan el acceso a contenido y la interacción, la investigación se alinea con el objetivo de optimizar estas herramientas mediante un análisis profundo de las causas de los problemas. Esto no solo incrementa la comprensión sobre los desafíos actuales, sino que también proporciona bases sólidas para el diseño de interfaces más funcionales.

* 1. **Población y muestra**

**Población:** La población, es el conjunto total de individuos, elementos o eventos que comparten una característica en común y que son el objeto de interés de una investigación. Este grupo es generalmente amplio y permite realizar inferencias basadas en estudios que lo representen [32].

**Muestra:** La muestra es un subconjunto representativo de la población, seleccionado de forma estratégica, con el propósito de recopilar datos que permitan estudiar las características de la población en su conjunto [32] .

**Población**

En esta investigación, la población corresponde a todos los estudiantes universitarios que interactúan activamente con plataformas educativas. Esta población se considera **infinita** debido a su extensión global y la imposibilidad de determinar un número exacto de estudiantes que utilizan estas plataformas.

**Muestra**

Por ser una población infinita, se procede al cálculo de la muestra utilizando la ecuación para poblaciones infinitas. Para este cálculo, se delimita la muestra a los estudiantes de software "A" pertenecientes a la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato en Ecuador, quienes interactúan con la plataforma educativa en cuestión.

La representatividad de esta muestra asegura que los resultados obtenidos sean válidos para el contexto general de estudiantes universitarios y puedan ser utilizados para cumplir con los objetivos de la investigación, particularmente en lo referente a la optimización de interfaces y la mejora de la experiencia del usuario.

1. **CAPITULO IV**

**MARCO ADMINISTRATIVO**

* 1. **Presupuesto**

**Recursos**

* **Humanos**  
    
  Dentro del presente proyecto los recursos humanos de los que se hará uso serán los investigadores a cargo Bryan Quitto, David Salinas, Maia Rojas y María Zapata además se contara con la opinión técnica de los usuarios que serán parte del muestreo, familiares directos de los investigadores y docentes de la Universidad.
* **Materiales**  
    
  Como recursos materiales se contará con una computadora personal para cada investigador, acceso a la plataforma educativa, servicio de internet y el repositorio bibliográfico de la universidad técnica de Ambato.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Costos | Directos | Categorías | Descripción | | Tipo de Recurso | Cantidad | Tipo de unidad | Valor unitario | Valor total |
| Materiales | Materias primas | |  |  |  |  |  |
| Materiales | | Tecnológico | 4 | Computadora | $500 | $2000 |
| Insumos | | Licencia | 1 | Licencia de Office por tres meses | $1,347.67 | $1,347.67 |
| Otros | |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Mano de obra directa | Desarrollador | | Humanos | 8 | Horas/Jornadas | $50 | $400 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Investigadores | | Humanos | 8 | Horas/Jornadas | $50 | $3200 |
|  |  |  |  |  |  |  |
| Indirectos | Costos indirectos de fabricación | Servicios | Internet | Tecnológico | 1 | Servicio mensual | $24.5 | $24.5 |
| Energía | Tecnológico | 1 | Servicio mensual | $30 | $30 |
| Agua |  |  |  |  |  |
| Otros |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| Gastos |  | Administración | | |  |  |  |  |  |
|  | Ventas | | |  |  |  |  |  |
|  | Otros | | | Alimentación | 4 | Alimentación de los investigadores | $5 | $20 |
|  |  |  |  |  |  |  |  |  | 5674.5 |

* 1. **Cronograma**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ACTIVIDADES** | **NOVIEMBRE** | | | | DICIEMBRE | | |
| SEM1 | SEM2 | SEM3 | SEM4 | SEM1 | SEM2 | SEM3 |
| **Introducción** |  |  |  |  |  |  |  |
| **Resumen** |  |  |  |  |  |  |  |
| **Palabras clave** |  |  |  |  |  |  |  |
| **1.       CAPÍTULO I: EL PROBLEMA** |  |  |  |  |  |  |  |
| **1.1. Tema** |  |  |  |  |  |  |  |
| **1.2. Planteamiento del problema: Maso** |  |  |  |  |  |  |  |
| **1.3. Planteamiento del problema: Meso** |  |  |  |  |  |  |  |
| **1.4. Planteamiento del problema: Micro** |  |  |  |  |  |  |  |
| **1.5. Análisis crítico: árbol de problemas** |  |  |  |  |  |  |  |
| **1.6. Análisis crítico: causa-efecto** |  |  |  |  |  |  |  |
| **1.7. Prognosis** |  |  |  |  |  |  |  |
| **1.8. Formulación del problema** |  |  |  |  |  |  |  |
| **1.9. Preguntas directrices: Variable independiente** |  |  |  |  |  |  |  |
| **1.10. Preguntas directrices: Variable dependiente** |  |  |  |  |  |  |  |
| **1.11. Preguntas directrices: Solución** |  |  |  |  |  |  |  |
| **1.12. Delimitación del objeto de la investigación** |  |  |  |  |  |  |  |
| **1.13. Justificación** |  |  |  |  |  |  |  |
| **1.14. Objetivos: Objetivo general** |  |  |  |  |  |  |  |
| **1.15. Objetivos: Objetivos específicos** |  |  |  |  |  |  |  |
| **2.       CAPÍTULO II: MARCO TEÓRICO** |  |  |  |  |  |  |  |
| **2.1. Investigaciones previas** |  |  |  |  |  |  |  |
| **2.2. Marco teórico: Constelación de ideas variable independiente** |  |  |  |  |  |  |  |
| **2.3. Marco teórico: Constelación de ideas variable dependiente** |  |  |  |  |  |  |  |
| **3.       CAPÍTULO III: METODOLOGÍA** |  |  |  |  |  |  |  |
| **3.1. Matriz metodológica** |  |  |  |  |  |  |  |
| **3.2. Nivel o tipo de investigación** |  |  |  |  |  |  |  |
| **3.3. Población y muestra** |  |  |  |  |  |  |  |
| **4.       CAPÍTULO IV: MARCO ADMINISTRATIVO** |  |  |  |  |  |  |  |
| **4.1. Presupuesto** |  |  |  |  |  |  |  |
| **4.2. Cronograma** |  |  |  |  |  |  |  |
| **4.3. Matriz de participación** |  |  |  |  |  |  |  |
| **4.4. Bibliografía** |  |  |  |  |  |  |  |
| **4.5. Anexos** |  |  |  |  |  |  |  |

* 1. **Matriz de participación**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Actividades** | **María Zapata** | **Maia Rojas** | **David Salinas** | **Bryan Quitto** |
| **Introducción** |  |  |  |  |
| **Resumen** |  |  |  |  |
| **Palabras clave** |  |  |  |  |
| **Tema** |  |  |  |  |
| **Planteamiento del problema: Maso** |  |  |  |  |
| **Planteamiento del problema: Meso** |  |  |  |  |
| **Planteamiento del problema: Micro** |  |  |  |  |
| **Análisis crítico: árbol de problemas** |  |  |  |  |
| **Análisis crítico: causa-efecto** |  |  |  |  |
| **Prognosis** |  |  |  |  |
| **Formulación del problema** |  |  |  |  |
| **Preguntas directrices: Variable independiente** |  |  |  |  |
| **Preguntas directrices: Variable dependiente** |  |  |  |  |
| **Preguntas directrices: Solución** |  |  |  |  |
| **Delimitación del objeto de la investigación** |  |  |  |  |
| **Justificación** |  |  |  |  |
| **Objetivos: Objetivo general** |  |  |  |  |
| **Objetivos: Objetivos específicos** |  |  |  |  |
| **Investigaciones previas** |  |  |  |  |
| **Marco teórico: Constelación de ideas variable independiente** |  |  |  |  |
| **Marco teórico: Constelación de ideas variable dependiente** |  |  |  |  |
| **Matriz metodológica** |  |  |  |  |
| **Nivel o tipo de investigación** |  |  |  |  |
| **Población y muestra** |  |  |  |  |
| **Presupuesto** |  |  |  |  |
| **Cronograma** |  |  |  |  |
| **Matriz de participación** |  |  |  |  |
| **Bibliografía** |  |  |  |  |
| **Anexos** |  |  |  |  |

**Bibliografía**