UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE CIENCIAS ECONÓMICAS Y DE ADMINISTRACIÓN

TRABAJO FINAL PARA OBTENER EL TÍTULO DE

MAGÍSTER EN SISTEMAS DE INFORMACIÓN DE LAS ORGANIZACIONES

Título

Impacto de la IA en la productividad de los equipos de desarrollo de software de la Intendencia de Montevideo

por

Martín Piperno
Gonzalo Martínez

TUTOR: María Corti

Montevideo
URUGUAY
2025

FCEA FACULTAD DE CIENCIAS ECONÓMICAS Y DE ADMINISTRACIÓN POSGRADOS | UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Página de Aprobación

El tribunal docente integrado por los abajo firmantes aprueba el Trabajo Final:

Título

……
……
…….....
.....

Autor/es

……
……

Tutor/Coordinador

……
……

Posgrado

……
……

Puntaje

……

Tribunal

Profesor……
……(nombre y firma).

Profesor……
……(nombre y firma).

Profesor……
……(nombre y firma).

FECHA.....

1 Resumen.....3
2 Introducción.....5
3 Hipótesis y Objetivos.....7
4 Estado del arte.....9
4.1 Herramientas de IA.....10
4.2 Riesgos asociados al uso de inteligencia artificial.....18
4.2.1 Riesgos de seguridad de la información.....18
4.2.2 Riesgos operativos y estratégicos.....20
4.2.3 Recomendaciones para mitigar riesgos.....22
4.3 Estrategias de gestión del cambio para la adopción de Inteligencia Artificial.....27
4.4 Casos de éxito.....30
5 Implementación.....31
5.1 Contexto Actual de la Intendencia de Montevideo.....31
5.2 Formas de usar la IA para mejorar la productividad.....37
5.3 Riesgos y medidas de mitigación.....45
5.3.1 Riesgos de seguridad de la información.....46
5.3.2 Riesgos operativos y estratégicos.....47
5.3.3 Recomendaciones estratégicas para la IM.....48
5.4 Plan incorporación de IA en los equipos de desarrollo.....49
5.4.1 Diagnóstico y Alcance del Piloto.....49
5.4.2 Selección y Configuración de Herramientas.....50
5.4.3 Capacitación y Sensibilización.....50
5.4.4 Ejecución del Piloto.....50
5.4.5 Evaluación de Resultados.....51
5.4.6 Recomendaciones y Escalado Progresivo.....51
6 Conclusiones.....53
7 Trabajos a Futuro.....56
8 Referencias bibliográficas y bibliografía.....58

1 Resumen

La presente tesis aborda la exploración y adopción de herramientas de inteligencia artificial (IA) en los equipos de desarrollo de software de la Intendencia de Montevideo, impulsada por el interés en optimizar la productividad y mantenerse a la vanguardia tecnológica. Dado el rápido avance tecnológico, es crucial para todas las instituciones incorporar soluciones innovadoras que mejoren sus procesos operativos.

Por otro lado, el estudio analiza cómo las herramientas basadas en IA pueden acelerar el aprendizaje de nuevas tecnologías y fortalecer el conocimiento sobre las ya existentes. A través de la automatización de tareas y el análisis de datos, estas herramientas tienen el potencial de optimizar procesos y aumentar la eficiencia. Sin embargo, su integración debe considerar cuidadosamente la seguridad y privacidad, con el fin de proteger la integridad de los sistemas y los datos.

También se evalúan los costos asociados con la implementación de IA, no sólo en términos de inversión inicial sino en relación con los beneficios a largo plazo. Un componente clave de la tesis es medir el impacto de la IA en la eficiencia operativa, para determinar si su adopción contribuye a los objetivos de la Intendencia y si puede extenderse a otras áreas de la administración pública.

El estudio contribuye al conocimiento sobre la integración de la IA en el desarrollo de software, abarcando temas como la seguridad de la información, la mejora de la productividad y la evaluación de costos y beneficios.

En términos metodológicos adoptaremos un enfoque mixto que combina métodos exploratorios, descriptivos, cuantitativos y cualitativos dentro de un marco no

experimental. A través de una búsqueda exhaustiva y el análisis de datos numéricos y cualitativos, se examina el impacto de estas herramientas, lo que permitirá a la Intendencia de Montevideo tomar decisiones informadas sobre su adopción. Además, se trata de un estudio de caso, aplicable a otras organizaciones.

Palabras Claves: IA, Intendencia de Montevideo, Herramientas de IA, Desarrollo de Software

2 Introducción

En los últimos años, la inteligencia artificial (IA) ha emergido como una de las tecnologías más transformadoras en el ámbito de la innovación digital. Su capacidad para automatizar tareas, procesar grandes volúmenes de información y asistir en la toma de decisiones ha captado el interés de organizaciones públicas y privadas que buscan mejorar sus niveles de eficiencia, calidad y adaptabilidad. Entre los diversos campos donde esta tecnología muestra un alto potencial de impacto, el desarrollo de software destaca por su dinamismo, complejidad técnica y necesidad constante de evolución.

En este contexto, la Intendencia de Montevideo ha manifestado un interés estratégico en explorar la aplicación de herramientas de IA para incrementar la productividad de sus equipos de desarrollo de software. Frente a una realidad marcada por la escasez de recursos humanos, la complejidad tecnológica y la demanda creciente de soluciones digitales eficientes, la integración de estas herramientas surge como una oportunidad para optimizar procesos clave, reducir tiempos de desarrollo y mejorar la calidad de los productos entregados.

Analizaremos de forma integral el impacto que puede tener la inteligencia artificial en los equipos de desarrollo de software de la Intendencia, abordando tanto sus beneficios como los riesgos asociados a su implementación. Se pone especial énfasis en dos dimensiones críticas: la seguridad de la información y la gestión del cambio organizacional, elementos fundamentales para garantizar una adopción responsable y sostenible.

La investigación se enmarca en un estudio de caso con enfoque metodológico mixto, que combina métodos exploratorios, descriptivos, cuantitativos y cualitativos. Esta aproximación permite no solo identificar las herramientas más adecuadas y las fases del

proceso de desarrollo susceptibles de ser optimizadas, sino también establecer métricas objetivas para evaluar su impacto real en términos de eficiencia y efectividad. Asimismo, se contemplan estrategias para gestionar el cambio cultural y técnico que implica la adopción de IA, considerando aspectos organizacionales, normativos y de capacitación.

En particular, se propone analizar cómo estas tecnologías pueden:

- Automatizar tareas repetitivas como la generación de código y pruebas.
- Agilizar el aprendizaje y la documentación técnica.
- Mejorar la calidad del software mediante sugerencias inteligentes.
- Reducir los tiempos de entrega (Lead Time), ciclo (Cycle Time) y trabajo en progreso (WIP).
- Asegurar la confidencialidad y la integridad de los datos institucionales.
- Facilitar la transición hacia una cultura de innovación y aprendizaje continuo.

3 Hipótesis y Objetivos

De la temática planteada en la introducción se deriva la siguiente hipótesis central de esta investigación:

La implementación de herramientas de inteligencia artificial en los equipos de desarrollo de software de la Intendencia de Montevideo incrementará su productividad, siempre que se gestione adecuadamente el cambio y se adopten medidas efectivas para proteger la seguridad de la información.

A partir de esta hipótesis, se establece el objetivo general que consiste en analizar cómo la inteligencia artificial puede mejorar la productividad de los equipos de desarrollo de software de la Intendencia de Montevideo, asegurando una gestión adecuada del cambio y la protección de la seguridad de la información.

Para alcanzar dicho propósito, se definen los siguientes objetivos específicos:

- Identificar las herramientas de IA más pertinentes para el contexto institucional.
- Evaluar su influencia sobre la eficiencia y efectividad del equipo de desarrollo.
- Establecer métricas para medir su impacto real.
- Determinar las fases del proceso de desarrollo más susceptibles de mejora mediante IA.
- Diseñar estrategias de cambio que aseguren una transición exitosa y sostenible.

Este trabajo aspira no solo a generar conocimiento académico sobre el uso de IA en el desarrollo de software, sino también a ofrecer una guía práctica que contribuya a la toma

de decisiones estratégicas dentro de la Intendencia de Montevideo y otras instituciones públicas con desafíos similares.

4 Estado del arte

Durante el desarrollo de esta tesis se llevó a cabo una revisión bibliográfica en diversos artículos y publicaciones académicas. Sin embargo, se identificó una dificultad recurrente: en el campo de la inteligencia artificial, los papers con varios años tienden a quedar obsoletos rápidamente, debido al ritmo acelerado de evolución tecnológica y la constante aparición de nuevos modelos, herramientas y marcos teóricos. A diferencia de otras disciplinas más estables, la IA requiere una actualización continua de fuentes para reflejar el estado actual de las prácticas y aplicaciones vigentes.

La inteligencia artificial tradicional se ha orientado históricamente hacia la resolución de problemas específicos mediante modelos predictivos y analíticos, como la clasificación, la regresión o la optimización. Su eficacia se basa en el análisis de datos históricos y en la aplicación de reglas predefinidas para obtener resultados concretos, lo cual ha demostrado ser útil en ámbitos como la detección de fraudes, el reconocimiento de patrones o la predicción de tendencias (Norvig & Russell, 2021). Sin embargo, esta aproximación presenta limitaciones cuando se requiere no solo analizar información, sino también generar nuevos contenidos que complementen el trabajo humano.

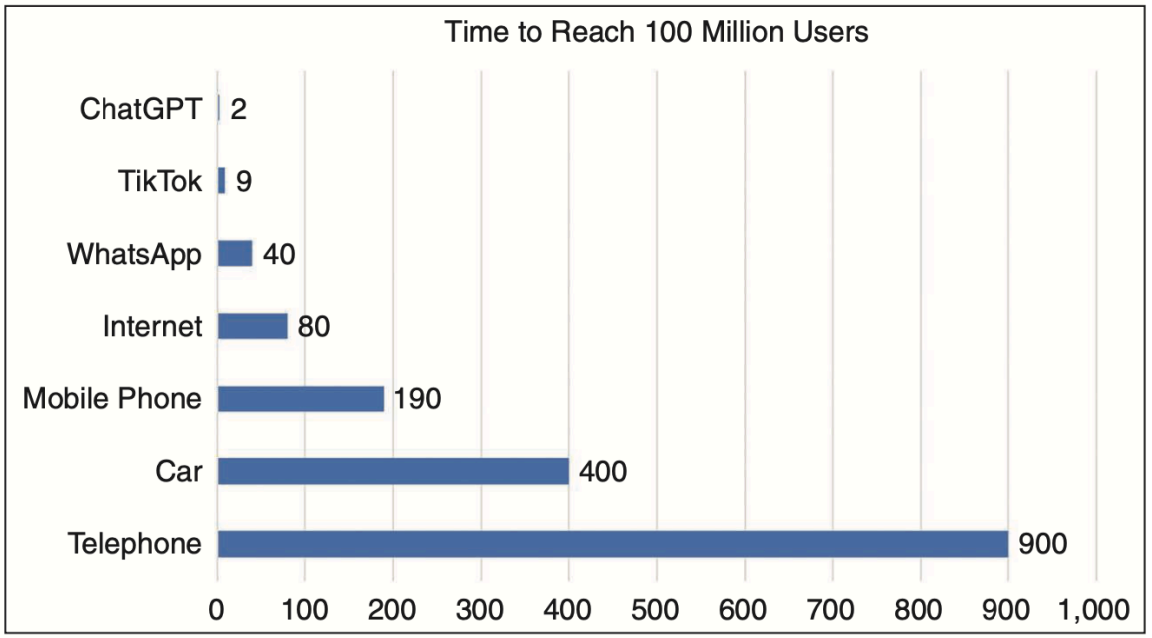
En contraste, la inteligencia artificial generativa representa una evolución significativa al enfocarse en la creación de resultados inéditos a partir de patrones aprendidos en grandes volúmenes de datos. Su desarrollo se ha potenciado con el uso de redes generativas adversarias (Goodfellow et al., 2014) y, más recientemente, con modelos basados en transformers que han ampliado las posibilidades de generación de texto, imágenes, código y otros formatos (Vaswani et al., 2017). Este tipo de inteligencia artificial se caracteriza por su capacidad de sintetizar información y producir artefactos nuevos que no estaban

explícitamente presentes en los datos de entrenamiento, lo que la convierte en una herramienta especialmente valiosa en contextos donde la creatividad y la generación de conocimiento son esenciales (Bommasani et al., 2021).

En el ámbito del desarrollo de software, la inteligencia artificial generativa ha demostrado un potencial destacado para asistir en la producción de código, pruebas automatizadas y documentación técnica. Estas capacidades no solo reducen tiempos de desarrollo, sino que también disminuyen la carga de tareas repetitivas que afectan la eficiencia de los equipos (NguyenDuc et al., 2023).

4.1 Herramientas de IA

Actualmente, existe un conjunto creciente de herramientas basadas en IA utilizadas en el desarrollo de software. Estas tecnologías permiten optimizar flujos de trabajo y mejorar la eficiencia en tareas como la generación de código, la depuración y la creación de documentación. A pesar de su efectividad en problemas sencillos y moderados, su desempeño disminuye ante desafíos más complejos, especialmente aquellos que requieren una interpretación detallada de los requisitos (Kuhail et al., 2024).



Tiempo medido en meses en alcanzar los 100 millones de usuarios en diferentes tecnologías (Ebert & Louridas, 2023).

Como se menciona en el artículo (Ebert & Louridas, 2023), antes de la popularización de herramientas de IA, los programadores dependían de recursos tradicionales como buscadores web y foros especializados (por ejemplo, Stack Overflow) para resolver problemas. Este enfoque, aunque efectivo, era más lento y dependía de la capacidad del programador para encontrar y adaptar soluciones existentes. Las herramientas de IA generativa han revolucionado este paradigma al ofrecer respuestas específicas y personalizadas en tiempo real, reduciendo significativamente la inversión de tiempo

Entre las herramientas que existen actualmente, algunas se encuentran más adaptadas que otras al ambiente de desarrollo de software. Entre ellas se encuentra el ChatGPT que ha sido ampliamente adoptado, aunque su implementación está mayormente enfocada en tareas específicas. Si bien se utilizan para mejorar la productividad, aún enfrentan limitaciones en términos de precisión y capacidad para entender contextos complejos. Estas herramientas han sido incorporadas en entornos de desarrollo integrados (IDEs)

como Visual Studio Code, GitHub Copilot y OpenAI Codex. Su incorporación está diseñada para complementar, no reemplazar, el trabajo del programador, especialmente en la creación de bloques de código estándar, generación de comentarios, y sugerencias para pruebas unitarias. Sin embargo, su implementación en proyectos a gran escala sigue siendo limitada debido a preocupaciones sobre su capacidad para manejar requisitos avanzados.

Name	URL	Technology	Cost	Use cases
ChatGPT	https://chat.openai.com/	GPT-4	USD$20 per month for the chat interface; pricing for application programming interface use depends on usage.	Code completion, code generation, code comprehension, reverse engineering, pseudo code reasoning and execution.
CoPilot	https://github.com/features/copilot	OpenAI CodeX, GPT-4	USD$10 per month/USD$100 per year for individuals, $19 per user per month for business plans.	Code completion for CoPilot; CoPilot X uses the more advanced GPT-4 model and can answer questions based on code documentation; aid in pull requests, shell commands, and scripting.
Tabnine	https://www.tabnine.com/	Proprietary ML engine trained with OSS	Basic tier is free, Pro tier starts at USD$12 month/user; also possible to self-host	Code completion. Runs also on private desktop to protect IPR.
Hugging Face (various different models)	https://huggingface.co/docs/transformers/index	Transformer, details vary depending on the model	Free and open source	Code completion and code generation, depending on the model.

Tabla tomada de (Ebert & Louridas, 2023) donde se comparan diferentes herramientas.

En el artículo (Kuhail et al., 2024), se realizó un estudio con 180 problemas de programación en LeetCode. Los resultados muestran que ChatGPT tiene un alto índice de éxito en problemas simples (86%) y moderados (70%), pero un rendimiento significativamente menor en problemas complejos (20%). Además, se citan casos específicos donde las herramientas han sido útiles para generar documentación o mejorar la calidad del código.

En el artículo de Frank & Godwin, 2024 se aborda el impacto de GitHub Copilot en la productividad de los desarrolladores de software. Su objetivo principal es analizar cómo las capacidades de autocompletado de código de esta herramienta afectan la eficiencia, calidad del código y experiencia del usuario, además de identificar sus beneficios y limitaciones.

El estudio utiliza un enfoque mixto que combina encuestas cuantitativas, para medir métricas de productividad, e investigaciones cualitativas, mediante entrevistas con desarrolladores de diferentes niveles de experiencia y áreas de especialización. Los participantes incluyen desde principiantes hasta profesionales avanzados, trabajando en dominios como desarrollo web y ciencia de datos.

Los hallazgos principales indican que GitHub Copilot mejora significativamente la productividad. Más del 78% de los participantes reportaron un aumento en su eficiencia, logrando completar tareas un 30% más rápido y ahorrando de dos a tres horas semanales en tareas repetitivas o de búsqueda de soluciones. La herramienta también contribuye a la calidad del código, ya que el 90% de los fragmentos generados fueron funcionalmente correctos. Sin embargo, se encontraron errores lógicos y de contexto en algunos casos, lo que resalta la necesidad de una supervisión humana constante.

En términos de satisfacción del usuario, el 85% de los desarrolladores expresaron una experiencia positiva con GitHub Copilot, destacando su capacidad para generar sugerencias contextuales y relevantes. Además, la personalización basada en los estilos de codificación individuales mejoró la percepción general de la herramienta. Por otro lado, se identificaron desafíos importantes, como errores ocasionales en las sugerencias generadas, que requerían ajustes manuales, y preocupaciones sobre la dependencia excesiva de la herramienta.

El documento también señala que GitHub Copilot presenta oportunidades educativas, permitiendo a los desarrolladores aprender nuevas técnicas y patrones de codificación. Sin embargo, existe preocupación sobre su impacto en el aprendizaje profundo de conceptos esenciales. Las conclusiones del estudio subrayan que, aunque GitHub Copilot ofrece ventajas significativas al permitir a los desarrolladores enfocarse en tareas de mayor nivel,

requiere un uso crítico para evitar errores y fomentar un equilibrio adecuado entre la automatización y la supervisión humana.

En términos de recomendaciones, el estudio sugiere mejorar las capacidades de personalización de la herramienta para adaptarse mejor a los estilos y niveles de habilidad de los usuarios, así como realizar estudios longitudinales para evaluar su impacto a largo plazo en las prácticas y habilidades de desarrollo. A futuro, herramientas como GitHub Copilot tienen el potencial de transformar el desarrollo de software, pero su implementación debe garantizar que los beneficios de la automatización no afecten la capacidad de los desarrolladores para mantener habilidades fundamentales y colaborar eficazmente en entornos profesionales.

Programación de pares vs persona-ia (pAIr)

En el paper (Ma et al., 2023) se describe que la programación en parejas entre humanos surgió en los años 90 como una práctica ágil para mejorar la calidad del código y fomentar el aprendizaje mutuo. Con la llegada de la IA, el enfoque ha cambiado hacia la automatización de tareas y la asistencia directa, reduciendo la dependencia de interacciones exclusivamente humanas. Se cuenta con estudios en donde compara la programación de a pares contra la programación con una IA. El enfoque está en comparar los beneficios, desafíos y diferencias entre la colaboración humano-humano y humano-IA, evaluando aspectos como calidad del código, productividad, aprendizaje y satisfacción

En el mismo artículo se analizan las similitudes y diferencias entre la programación en parejas humano-humano y la programación colaborativa humano-IA, enfocándose en herramientas como GitHub Copilot. En el caso de las parejas humanas, la colaboración mejora la calidad del código, la productividad y el aprendizaje, aunque puede ser costosa en términos de tiempo y logística. Por otro lado, las parejas humano-IA prometen ventajas

como la personalización de la asistencia según el nivel de habilidad del usuario, el soporte en tareas repetitivas y la eliminación de problemas organizativos. Sin embargo, estas herramientas enfrentan desafíos en tareas complejas, donde la calidad y confiabilidad del código generado pueden ser insuficientes.

El éxito de la programación en pareja depende de factores como la complejidad de las tareas, la compatibilidad entre los participantes y la calidad de la comunicación. En el caso de la programación humano-IA, se observa que las herramientas deben adaptarse al nivel del usuario y fomentar interacciones más productivas para compensar la falta de comunicación natural. En el ámbito educativo, las IA pueden ser útiles para guiar a estudiantes principiantes, generar materiales de aprendizaje y proporcionar retroalimentación, pero también se corre el riesgo de fomentar la dependencia excesiva de estas herramientas, lo que podría limitar el desarrollo de habilidades fundamentales.

El documento también aborda preocupaciones éticas y sociales, como la responsabilidad en el uso del código generado y el impacto de la IA en las dinámicas laborales y educativas. Aunque la programación humano-IA presenta oportunidades significativas, como la capacidad de personalizar las tareas y fomentar la colaboración efectiva, requiere superar retos técnicos, éticos y pedagógicos para maximizar su potencial. Este análisis concluye que, si bien la IA no reemplaza completamente la interacción humana, puede redefinir la forma en que se realizan las colaboraciones en el desarrollo de software y la educación.

Moderating Factors	Human-Human Challenges	Human-AI Opportunities
Task Types & Complexity: pair work better if the task is not too simple and good for collaboration [16, 76]	Hard to design suitable tasks of appropriate complexity level	AI may be used to generate collaboration tasks and adjust tasks complexity
Compatibility: pairs with similar skill levels and compatible working styles work better [16, 70]	Hard to find a similarly skilled or compatible partner	AI partner should adjust to human skill level and adapt to be compatible with different people
Communication: pairs work better with productive conversations [24], and critiques lead to more debugging success [55]	Hard to teach effective communication and constructive criticism	AI partner should support productive conversations and provide critiques
Collaboration: pairs work better with positive interdependence [67] and clear and balanced responsibilities [57]	Hard to teach collaboration and prevent free riders	AI should support positive social interactions and collaboration and avoid over-assist that eliminates human's need to engage
Logistics: pair programming is costly to implement because of management challenges [4, 79]	Hard to schedule and assess individual contributions in a pair	Scheduling is no longer a problem, but humans should be accountable and responsible when using AI-generated code

Tabla tomada del artículo (Ma et al., 2023) se resumen los desafíos de la programación Humano-Humano que presentan oportunidades para la IA.

Desafíos

El artículo (Mbizo et al., 2024) identifica varios desafíos éticos y técnicos relacionados con la integración de herramientas de IA generativa en el desarrollo de software. En términos éticos, uno de los principales problemas es la privacidad y seguridad de los datos, ya que estas herramientas procesan grandes volúmenes de información, lo que puede comprometer datos sensibles si no se manejan correctamente. Además, existe el riesgo de una excesiva dependencia de las herramientas de IA, lo que podría erosionar la capacidad crítica y la responsabilidad humana en las decisiones técnicas. También se destacan implicaciones en la equidad y accesibilidad, ya que no todas las organizaciones o individuos tienen acceso a estas tecnologías, lo que podría acentuar las desigualdades existentes. En el ámbito educativo, el uso de la IA generativa puede limitar el desarrollo de competencias fundamentales en los estudiantes, generando una dependencia que podría afectar su capacidad para resolver problemas por sí mismos.

En cuanto a los desafíos técnicos, las herramientas de IA enfrentan problemas relacionados con la fiabilidad y la precisión, ya que a menudo generan errores o resultados incorrectos que requieren supervisión y corrección humanas. Asimismo, estas herramientas tienen

dificultades para interpretar contextos complejos o entender requisitos específicos de los proyectos. Otro reto es la falta de personalización, ya que adaptar las herramientas al nivel de habilidad de los desarrolladores, especialmente de los principiantes, sigue siendo una tarea compleja. Además, la integración de estas tecnologías en equipos diversos puede ser complicada debido a la incompatibilidad con distintos estilos de programación y dinámicas de colaboración.

Estos desafíos destacan la necesidad de equilibrar los beneficios de la IA generativa con una implementación ética y técnicamente sólida, que minimice riesgos y maximice su utilidad tanto en la industria como en la educación.

Visión de Futuro

Según lo que se puede obtener de los diferentes artículos analizados anteriormente, estas herramientas evolucionarán hacia una mayor precisión y sofisticación, incluyendo la capacidad de entender mejor los requisitos del usuario y ofrecer soluciones más completas. Sin embargo, existe una preocupación latente entre los programadores de que estas tecnologías podrían desplazar roles de nivel básico, limitando oportunidades para desarrolladores novatos. En un escenario ideal, la IA se integraría como una asistente robusta en lugar de un reemplazo completo, equilibrando innovación con la preservación de habilidades humanas esenciales. Por otro lado, la mayoría de las empresas adoptarán estrategias de desarrollo y pruebas aumentadas por IA. Además, se espera que las herramientas de IA generativa evolucionen, permitiendo aplicaciones más sofisticadas, aunque con desafíos éticos y de seguridad que requerirán supervisión humana constante.

Se prevé que la dependencia de estas herramientas podría modificar las habilidades esenciales necesarias en los programadores, generando una menor autonomía y capacidad crítica en los procesos de desarrollo. Además, la integración ética de estas tecnologías será

crucial para garantizar que su adopción no perpetúe desigualdades o comprometa la privacidad y seguridad de los datos. A nivel organizacional, las empresas necesitarán equilibrar la automatización ofrecida por la IA con la supervisión humana para mantener la calidad y la coherencia de los proyectos.

4.2 Riesgos asociados al uso de inteligencia artificial

Como se mencionó anteriormente, la incorporación de herramientas de inteligencia artificial en el desarrollo de software representa una oportunidad significativa para mejorar la productividad, optimizar procesos y elevar la calidad del código. No obstante, su implementación también introduce una serie de riesgos que deben ser identificados, analizados y gestionados adecuadamente, en particular aquellos que afectan la seguridad de la información, la privacidad, la integridad institucional y la calidad técnica de los sistemas.

Seguidamente, se presentan los principales riesgos asociados a la adopción de herramientas de IA, distinguiendo entre aquellos que comprometen la seguridad de la información y los que suponen riesgos operativos o estratégicos.

4.2.1 Riesgos de seguridad de la información

Dentro de los riesgos asociados a la seguridad de la información se presentan riesgos de acceso a información sensible, riesgos normativos y de seguridad del código, los mismo se detallan a continuación.

Fugas de información confidencial o sensible

Las herramientas de IA, como GitHub Copilot, Amazon CodeWhisperer o ChatGPT, requieren procesar datos ingresados por los usuarios. Esto puede implicar,

inadvertidamente, el envío de información sensible a servidores externos, como fragmentos de código fuente, credenciales, tokens de acceso o datos personales. Dado que muchos de estos servicios operan en la nube, los datos ingresados pueden ser almacenados o utilizados para mejorar los modelos subyacentes, lo que incrementa el riesgo de fugas de información fuera del entorno institucional (Pearce et al., 2022).

Exposición de datos personales y riesgos normativos

El tratamiento de información personal o institucional por parte de herramientas IA en la nube puede derivar en incumplimientos normativos. En Uruguay, la Ley N.º 18.331 de Protección de Datos Personales impone restricciones sobre la transferencia internacional de datos y exige garantías de confidencialidad (IMPO, 2008). Asimismo, en interacciones con actores internacionales, deben considerarse regulaciones como el Reglamento General de Protección de Datos (GDPR) de la Unión Europea (European Union, 2016). El uso no regulado de IA puede contravenir estas normativas y derivar en sanciones legales o reputacionales.

Introducción de vulnerabilidades de seguridad en el código

Las herramientas de generación automática de código pueden producir fragmentos que, aunque funcionales, no contemplan buenas prácticas de seguridad. Se han documentado casos donde asistentes como Copilot generaron código vulnerable a inyecciones SQL, cross-site scripting (XSS) o manejo inseguro de excepciones (Pearce et al., 2022). La omisión de una revisión humana rigurosa puede facilitar la introducción de estas vulnerabilidades en entornos de producción (*A03 Injection*, n.d.).

Ataques a través de plataformas de IA

Las soluciones de IA generativa pueden ser explotadas como vectores de ataque. Uno de los métodos emergentes es la **inyección de instrucciones (prompt injection)**, en la cual un atacante incorpora comandos ocultos o maliciosos en una entrada aparentemente inofensiva. Este tipo de ataque puede engañar al modelo para que revele información sensible o ejecute acciones no previstas (Microsoft, 2023; *OWASP AI Security and Privacy Guide*, n.d.).

4.2.2 Riesgos operativos y estratégicos

Además de los riesgos de la seguridad de la información, también existen riesgos operativos y estratégicos asociados.

Decisiones erróneas o sesgadas basadas en IA

Los modelos de inteligencia artificial pueden estar entrenados con datos sesgados, incompletos o no representativos del contexto específico en el que se aplican (Pearce et al., 2022). Esta limitación puede generar recomendaciones erróneas o decisiones inadecuadas. En el ámbito del desarrollo de software, esto se traduce en pruebas automatizadas que no contemplan casos críticos, sugerencias de código inseguras o decisiones arquitectónicas inadecuadas.

Por ejemplo, herramientas como GitHub Copilot han demostrado en estudios recientes que pueden sugerir patrones de programación inseguros, especialmente cuando los modelos subyacentes fueron entrenados con código fuente público que incluye prácticas de seguridad desactualizadas o incorrectas. Este tipo de sesgo puede introducir vulnerabilidades en el sistema si no se realiza una revisión humana exhaustiva.

Dependencia excesiva y pérdida de conocimiento interno

Una adopción acrítica de herramientas de IA puede derivar en una dependencia tecnológica que afecte la autonomía operativa del equipo. Si los desarrolladores utilizan sistemáticamente asistentes sin comprender los fundamentos del código generado, se debilita el conocimiento técnico interno, dificultando la resolución de problemas complejos sin la asistencia de terceros (NIST, 2023).

Disminución del control sobre la calidad del desarrollo

La automatización de procesos clave como la generación o revisión de código puede debilitar el control de calidad si no se establecen mecanismos de validación adecuados. En ausencia de supervisión humana, los sistemas podrían incorporar malas prácticas de programación, baja legibilidad o diseños ineficientes (*A03 Injection*, n.d.).

Manejo del Copyright y derechos de autor

Otros puntos a considerar al momento de incorporar IA para la generación de código son los vinculados a la propiedad intelectual y el copyright. Estas herramientas, entrenadas con grandes volúmenes de datos y código fuente disponible públicamente, pueden generar contenido que, en algunos casos, reproduce fragmentos protegidos por derechos de autor sin que el usuario sea plenamente consciente de ello (Pearce et al., 2022; Bommasani et al., 2021).

Una de las principales preocupaciones surge con asistentes de codificación como GitHub Copilot, los cuales han sido objeto de cuestionamientos legales al generar código que podría estar basado en repositorios bajo licencias restrictivas. En estudios recientes, se han identificado casos donde fragmentos generados coincidían parcialmente con código previamente publicado bajo licencias de tipo GPL, MIT u otras, lo que podría dar lugar a

conflictos de copyright si se reutilizan sin atribución o sin respetar sus términos (Pearce et al., 2022).

Asimismo, el uso de modelos de lenguaje como ChatGPT, que pueden generar textos completos, documentación o incluso fragmentos de código, no garantiza la originalidad del contenido generado, dado que el modelo podría reproducir, de forma involuntaria, partes del material en el que fue entrenado.

A nivel organizacional, se sugiere establecer políticas institucionales claras sobre el uso responsable y ético de la inteligencia artificial en entornos de desarrollo, definiendo procedimientos, roles y responsabilidades. Finalmente, resulta clave documentar adecuadamente el origen de los fragmentos generados, incluyendo la herramienta utilizada y el contexto en que fueron creados, a fin de garantizar trazabilidad y resguardar la integridad jurídica de los proyectos.

4.2.3 Recomendaciones para mitigar riesgos

A fin de garantizar una adopción responsable y segura de herramientas de inteligencia artificial en entornos de desarrollo de software institucional, resulta indispensable establecer una serie de medidas preventivas y correctivas que permitan minimizar los riesgos identificados previamente. Estas recomendaciones deben formar parte de una estrategia integral que abarque la gobernanza tecnológica, la capacitación de los equipos, el cumplimiento normativo, la gestión operativa y la supervisión continua. A continuación, se desarrollan cinco líneas de acción prioritarias que la organización debería implementar.

Políticas institucionales claras sobre el uso de herramientas de IA

Es fundamental establecer políticas institucionales claras sobre el uso de herramientas de inteligencia artificial. Estas políticas deben definir con precisión qué tipo de información

puede ser compartida con herramientas externas, cuáles son los límites en relación al tratamiento de datos personales o confidenciales, y qué procedimientos deben seguirse para garantizar la protección de la información. Además, deben contemplar criterios para la autorización de herramientas específicas, las condiciones de uso por parte del personal, y las responsabilidades asociadas a su aplicación. La existencia de un marco normativo interno robusto proporciona certeza jurídica, favorece la toma de decisiones responsables y establece las bases para una cultura institucional de uso ético y seguro de la tecnología (European Union, 2016; IMPO, 2008).

Preferencia por soluciones empresariales de IA

En entornos institucionales, especialmente en el sector público, se recomienda priorizar el uso de herramientas de inteligencia artificial en sus versiones empresariales. Estas versiones están diseñadas para cumplir con estándares más estrictos en materia de seguridad, privacidad y cumplimiento normativo, lo que las hace más adecuadas para el tratamiento de información sensible. A diferencia de las versiones de uso general, las soluciones empresariales permiten configurar políticas avanzadas de acceso, implementar controles de seguridad personalizados, realizar auditorías detalladas y gestionar el ciclo de vida de los datos conforme a marcos regulatorios locales e internacionales.

Por ejemplo, plataformas como Azure OpenAI Service, Google Cloud Vertex AI o AWS Bedrock ofrecen versiones empresariales de modelos de lenguaje (como GPT-4, PaLM o Claude) con opciones de privacidad reforzada, cifrado de extremo a extremo, control sobre el almacenamiento de datos y garantías contractuales de no utilización del contenido ingresado para reentrenar modelos. Estas condiciones son fundamentales para cumplir con regulaciones como el Reglamento General de Protección de Datos (GDPR) en Europa o la Ley N.º 18.331 de Protección de Datos Personales en Uruguay.

Según el AI Risk Management Framework del NIST (2023), el uso de herramientas empresariales reduce significativamente el riesgo de exposición no deseada de datos, facilita la trazabilidad de decisiones automatizadas y mejora la capacidad de las organizaciones para aplicar principios como la responsabilidad algorítmica, la transparencia y la mitigación de sesgos. En organismos públicos, donde se gestiona información ciudadana y servicios críticos, estas garantías no sólo son deseables, sino legal y éticamente obligatorias.

Capacitación en uso ético, seguro y legal de la IA

Es importante que el equipo reciba formación para usar herramientas de inteligencia artificial de forma ética, segura y conforme a la normativa vigente. Esta capacitación debe incluir tanto aspectos técnicos como legales, considerando el uso adecuado de asistentes de codificación, la validación de resultados y la detección de sesgos, especialmente en lo referido a la protección de datos personales. Estudios recientes advierten sobre errores y sesgos en los modelos de IA, lo que exige una supervisión crítica por parte de los equipos humanos (Pearce et al., 2022).

Supervisión humana en etapas críticas del desarrollo

Una recomendación fundamental consiste en establecer procesos de supervisión humana obligatoria en todas las etapas críticas del desarrollo asistido por inteligencia artificial. Aunque las herramientas generativas pueden acelerar ciertas tareas, su uso no exime de la revisión profesional de los productos generados, especialmente cuando se trata de código fuente, análisis de datos o decisiones automatizadas que puedan afectar a personas o servicios. Tal como lo señala OWASP (*A03 Injection*, n.d.), es frecuente que los modelos generativos sugieran código funcional pero inseguro si no se aplican mecanismos de validación. La intervención humana permite identificar errores, mitigar sesgos, validar la

coherencia de los resultados y garantizar que los sistemas mantengan la calidad y la seguridad requeridas. Esta práctica también contribuye a preservar el conocimiento técnico dentro de la organización y a mantener el control sobre las decisiones clave en los procesos de desarrollo (NIST, 2023).

Auditorías periódicas integradas al sistema de gestión institucional

Finalmente, se recomienda implementar auditorías periódicas sobre el uso de herramientas de inteligencia artificial, integradas al sistema institucional de gestión de calidad y de seguridad de la información. Estas auditorías deben evaluar el cumplimiento de las políticas internas, identificar posibles desviaciones, y proponer mejoras continuas. Asimismo, deben incluir la revisión de los registros de uso de IA, el análisis de los riesgos emergentes y la verificación del cumplimiento normativo, tanto a nivel nacional como internacional (Microsoft, 2023; *OWASP AI Security and Privacy Guide*, n.d.). La integración de estas auditorías al ciclo de mejora institucional permitirá monitorear de forma proactiva el impacto de la IA en los procesos de trabajo y garantizar que su uso se mantenga dentro de los límites éticos, legales y operativos definidos por la organización.

En conjunto, estas recomendaciones buscan sentar las bases para una adopción responsable y sostenible de la inteligencia artificial en el ámbito institucional, permitiendo aprovechar sus beneficios sin comprometer la seguridad de la información, la calidad de los servicios ni la confianza de la ciudadanía.

A continuación, se incluye una tabla que sintetiza los principales aspectos de la sección.

Categoría	Riesgo identificado	Descripción	Mitigaciones recomendadas
Seguridad de la información	Fugas de información confidencial o sensible	Posible exposición de código, credenciales o datos al usar herramientas de IA en la nube.	Políticas internas claras, anonimización de datos, uso de versiones empresariales de IA.
	Exposición de datos personales y riesgos normativos	Incumplimiento de la Ley 18.331 en Uruguay o del GDPR en Europa por transferencia y tratamiento inadecuado de datos.	Controles de acceso, monitoreo de uso, capacitación en normativa, auditoría y trazabilidad.
	Introducción de vulnerabilidades en el código	Generación de código inseguro (SQL injection, XSS, etc.) por asistentes como Copilot.	Revisiones humanas obligatorias, pruebas automatizadas de seguridad, estándares OWASP.
	Ataques mediante IA (prompt injection)	Manipulación de entradas para obtener información o ejecutar acciones no autorizadas.	Sanitización de entradas, filtros de seguridad, auditorías periódicas de interacciones.
Operativos y estratégicos	Decisiones erróneas o sesgadas	Modelos entrenados con datos incompletos pueden generar pruebas o código inadecuado.	Validación humana, datasets locales, capacitación en detección de sesgos.
	Dependencia excesiva y pérdida de conocimiento interno	Riesgo de pérdida de expertise si se delega demasiado en la IA.	Programación en pares, equilibrio entre IA y revisiones manuales, documentación interna.
	Disminución del control de calidad	La automatización sin controles adecuados puede introducir malas prácticas de desarrollo.	Estándares de calidad, auditorías de código, validación en procesos CI/CD.
	Manejo del copyright y derechos de autor	Generación de código basado en repositorios con licencias restrictivas.	Registro de origen del código, revisión legal, capacitación en propiedad intelectual.

Tabla que resume los riesgos y las medidas de mitigación.

4.3 Estrategias de gestión del cambio para la adopción de Inteligencia Artificial

Desarrollar estrategias para gestionar el cambio en el proceso operativo de desarrollo de software, asegurando una transición exitosa y un impacto positivo con la adopción de inteligencia artificial es un aspecto crucial para el equipo de desarrollo de software de la Intendencia de Montevideo. La incorporación de herramientas de inteligencia artificial (IA) implica cambios significativos en la dinámica habitual, por lo que es fundamental contar con una estrategia bien definida que facilite no sólo la aceptación de estas tecnologías, sino también una transición efectiva hacia una mayor productividad y satisfacción laboral (Kotter, 2012; Hiatt, 2006).

Comunicación Efectiva del Cambio

Una comunicación efectiva del cambio es vital para asegurar una transición exitosa hacia el uso de la IA. Comunicar claramente los objetivos, beneficios y desafíos asociados a estas tecnologías, explicando cómo la IA optimizará las tareas diarias al complementar las capacidades del equipo humano, reducirá resistencias iniciales (Kotter & Cohen, 2002). Por ejemplo, generar espacios regulares para intercambiar opiniones con los integrantes del equipo permitirá abordar preocupaciones específicas y recoger retroalimentación valiosa para ajustar oportunamente la implementación.

Diagnóstico y Selección de Herramientas Adecuadas

Realizar un diagnóstico preciso y seleccionar herramientas adecuadas es esencial antes de implementar la IA. Un análisis profundo de las necesidades del equipo y de las plataformas tecnológicas existentes facilitará elegir soluciones específicas que se integren fácilmente en los flujos actuales de trabajo (Cameron & Green, 2015). Implementar inicialmente

proyectos piloto para evaluar la efectividad de cada herramienta minimizará riesgos técnicos y operativos.

Para una implementación efectiva de la IA, es recomendable iniciar con proyectos piloto de dimensión acotada y criticidad media o baja. Esto implica seleccionar casos de uso que sean representativos pero no esenciales para las operaciones críticas del negocio, lo que permite evaluar la viabilidad de la solución sin comprometer procesos clave. Es recomendable comenzar con pilotos de tamaño pequeño a mediano, involucrando equipos reducidos y procesos limitados. Esto facilita una gestión más sencilla, reduce la complejidad y permite una evaluación más precisa de los resultados, además de facilitar ajustes rápidos antes de una implementación a gran escala.

Los proyectos piloto deben enfocarse en áreas de baja a media criticidad, evitando procesos esenciales cuya falla podría tener consecuencias graves. Esto permite identificar y mitigar riesgos sin afectar operaciones críticas.

En resumen, iniciar con proyectos piloto de IA de dimensión acotada y criticidad media o baja permite una evaluación controlada y efectiva de la tecnología, facilitando su integración exitosa en la organización.

Capacitación y Acompañamiento Continuo

La capacitación continua y el acompañamiento adecuado resultan fundamentales en la adopción exitosa de nuevas tecnologías (Hayes, 2018). Esto implica brindar formación específica y constante en el uso práctico de herramientas como asistentes de código, generadores automáticos de documentación o plataformas de automatización de pruebas. Además, la creación de un grupo interno especializado en IA facilitará la difusión de conocimientos y apoyo constante al resto del equipo.

Integración Gradual y Controlada de la IA

La integración gradual y controlada de la IA en procesos críticos es otro aspecto clave. Inicialmente, debería aplicarse en tareas menos críticas o más rutinarias, como generación inicial de documentación técnica, automatización de tareas administrativas o generación automática de casos de prueba, para luego expandirse gradualmente a funciones más complejas bajo estricta supervisión humana (Kotter, 2012).

Gestión del Cambio y Acompañamiento Interno

Un adecuado acompañamiento y gestión del cambio interno serán esenciales para minimizar resistencias y garantizar aceptación. Establecer líderes internos para guiar la transición, comunicar regularmente avances y fomentar espacios para compartir experiencias promoverá un ambiente positivo hacia el cambio tecnológico (Hiatt, 2006).

Medición Constante del Impacto y Ajuste Continuo

La medición constante del impacto y los ajustes continuos asegurarán que la incorporación de IA genere resultados concretos y positivos. Establecer desde el inicio métricas claras permitirá evaluar efectivamente la contribución de la IA e identificar rápidamente áreas que requieran ajustes adicionales (Anderson & Anderson, 2010).

Promoción de una Cultura de Innovación y Aprendizaje Permanente

Finalmente, promover una cultura de innovación y aprendizaje permanente dentro de la organización asegurará que la adopción de IA sea sostenible en el tiempo. Incentivar la experimentación y reconocer propuestas innovadoras facilitará una mejora continua en la productividad, eficiencia y calidad de los desarrollos de software (Hayes, 2018; Cameron & Green, 2015).

4.4 Casos de éxito

Diversas organizaciones internacionales han comenzado a incorporar herramientas de inteligencia artificial en sus procesos de desarrollo de software, con estrategias que combinan innovación y control de riesgos. Citigroup (Ashare, 2025), por ejemplo, aplicó IA generativa para apoyar a más de 30.000 desarrolladores en la modernización de sistemas heredados, facilitando la transformación de código antiguo y la automatización de tareas de documentación y análisis, como parte de un programa de transformación digital corporativo. En la misma línea, Accenture (Accenture, n.d.) adoptó GitHub Copilot de manera controlada, iniciando su implementación en equipos de automatización de pruebas y únicamente después de evaluar cuidadosamente aspectos de privacidad y licencias, utilizando entornos de desarrollo cerrados para mitigar riesgos legales.

Por su parte, SAP (Responsible AI, n.d.) diseñó un programa integral de "IA Responsable", que incluyó formación obligatoria para los equipos de desarrollo en temas de propiedad intelectual, privacidad y uso ético de modelos generativos. Microsoft (Taylor, 2025), a través del equipo de Azure, enfocó la aplicación de IA en la optimización de documentación técnica y en la automatización de respuestas en foros internos, logrando reducir en un 50% el tiempo promedio de resolución de tickets en tan solo tres meses, según reportes de DevOps. Finalmente, Uber (Fu & Soman, 2021) implementó IA para el análisis automático de logs y trazas de errores, destacando mejoras significativas en la velocidad de diagnóstico, aunque también identificando preocupaciones vinculadas a la confianza en los resultados, lo que motivó la incorporación de validaciones cruzadas con intervención humana como parte del flujo estándar de trabajo.

5 Implementación

La implementación de inteligencia artificial (IA) en el desarrollo de software representa una oportunidad transformadora para mejorar la productividad, eficiencia y calidad de los equipos técnicos. Este capítulo explora cómo las tecnologías basadas en IA pueden integrarse de manera efectiva en los flujos de trabajo de desarrollo, abordando tanto los beneficios como los desafíos que esta transformación implica, particularmente en la incorporación de estas tecnologías en la Intendencia de Montevideo. Como primer paso se realizó una entrevista con referentes de la Intendencia de Montevideo para conocer su situación actual y qué problemáticas son las que enfrentan.

5.1 Contexto Actual de la Intendencia de Montevideo

El equipo de desarrollo de software objeto de esta investigación está conformado por aproximadamente 70 funcionarios, que sumado al personal proveniente de proveedores externos, alcanza una cifra cercana a las 90 personas. Históricamente, este equipo ha presentado una baja rotación, limitando así la incorporación de nuevos integrantes. Esto ha generado un entorno consolidado y experimentado que maneja una complejidad considerable debido a la diversidad de roles y tecnologías involucradas en sus operaciones y proyectos.

Infraestructura tecnológica y uso actual

El equipo utiliza una infraestructura tecnológica diversa, destacándose plataformas como SAP, gestionadas por un grupo específico de informáticos junto a analistas funcionales, y tecnologías ampliamente adoptadas como Java, Oracle Forms y Reports, y PL/SQL.

Adicionalmente, Lotus Domino es empleado por dos miembros del equipo dedicados exclusivamente a esta tecnología.

En el ámbito del análisis de datos, el equipo utiliza diversas herramientas especializadas, mientras que para el desarrollo web se apoya en plataformas y lenguajes orientados a la creación de portales y aplicaciones móviles. La infraestructura tecnológica se basa en servidores de aplicaciones y entornos empresariales robustos, junto con frameworks frontend modernos. Asimismo, la implementación de microservicios y contenedores desempeña un papel relevante para gestionar y desplegar un gran número de aplicaciones, actualmente cuentan con más de 150. La gestión del código fuente y los entornos de desarrollo se lleva a cabo mediante soluciones de control de versiones y herramientas de DevOps.

Uso actual y potencial de Inteligencia Artificial

Actualmente, el equipo ha comenzado a explorar el uso de herramientas de inteligencia artificial, destacándose la utilización inicial de ChatGPT, asistentes como GitHub Copilot, así como plataformas emergentes como Cursor y Devin, aunque la adopción todavía es limitada y no se encuentra completamente integrada en los flujos de trabajo operativos. Las principales aplicaciones identificadas hasta ahora incluyen la asistencia en resolución de problemas y bugs, generación automática de casos de prueba, aprendizaje acelerado de nuevas tecnologías, automatización de tareas repetitivas, generación y documentación automática de código, y diseño preliminar de interfaces y formularios.

Cursor es un editor de código impulsado por inteligencia artificial, desarrollado por Anysphere. Basado en Visual Studio Code, ofrece características como autocompletado inteligente, reescrituras de código y comprensión de la base de código mediante

instrucciones en lenguaje natural. Su modo "agente" permite ejecutar tareas de desarrollo completas, mejorando la productividad de los desarrolladores.

Devin es una herramienta de asistencia digital autónoma creada por Cognition Labs. Diseñada para completar tareas de desarrollo de software, Devin puede programar, depurar, planificar y resolver problemas mediante técnicas de aprendizaje automático. Su capacidad para buscar recursos en línea y ajustar planes en tiempo real la convierte en una herramienta poderosa para desarrolladores que buscan automatizar procesos complejos.

Pese a estos primeros acercamientos, el uso sistemático y estándar de la inteligencia artificial aún es incipiente dentro del equipo, siendo utilizado solo por algunos miembros que han comenzado a explorar estas tecnologías de forma experimental.

Estructura organizacional y roles

El equipo está estructurado en tres grandes áreas: Aplicaciones (desarrollo y productos), Infraestructura (que abarca administración de sistemas, operaciones, seguridad, administración de base de datos y telecomunicaciones) y Relacionamiento Interno. Esta última área incluye Asistencia Técnica, que da soporte y asesora a quienes requieren servicios de tecnología de información; Mesa de Servicios, encargada de recepcionar y gestionar las incidencias y solicitudes de servicio realizadas por los usuarios de sistemas informáticos; y Análisis de Procedimientos Informáticos, que identifica oportunidades de mejora en los servicios y analiza los requerimientos.

En el área de Aplicaciones, se realizan tanto desarrollos propios como configuraciones y personalizaciones de productos existentes.

La mayoría de los miembros del equipo poseen un perfil senior con una extensa experiencia acumulada dentro de la organización, lo cual representa una fortaleza

significativa en términos de conocimiento interno, pero también plantea desafíos en la incorporación y adaptación a nuevas tecnologías.

Modalidad de trabajo y metodologías utilizadas

Actualmente, la modalidad de trabajo predominante es mixta, con mayor énfasis en el teletrabajo y asistencia presencial ocasional, dependiendo del grupo específico al que pertenezca cada integrante del equipo. El equipo está organizado en 12 grupos operativos, que varían en tamaño desde 1 hasta 8 miembros.

En términos de metodologías, aunque recientemente se ha impulsado la adopción gradual de enfoques ágiles como Scrum o Kanban, todavía prevalecen metodologías más tradicionales en la mayoría de los procesos operativos, reflejando una transición gradual y un desafío cultural significativo hacia modelos más adaptativos e innovadores.

Métricas utilizadas actualmente

Dentro de los proyectos que utilizan metodologías ágiles se comenzó a utilizar los conceptos de *Lead Time*, *Cycle Time* y *WIP* (Trabajo en Progreso) como métricas clave para medir la eficiencia de los equipos de desarrollo. Estas pueden ser utilizadas para evaluar el impacto del uso de la IA en los proyectos de software de la intendencia.

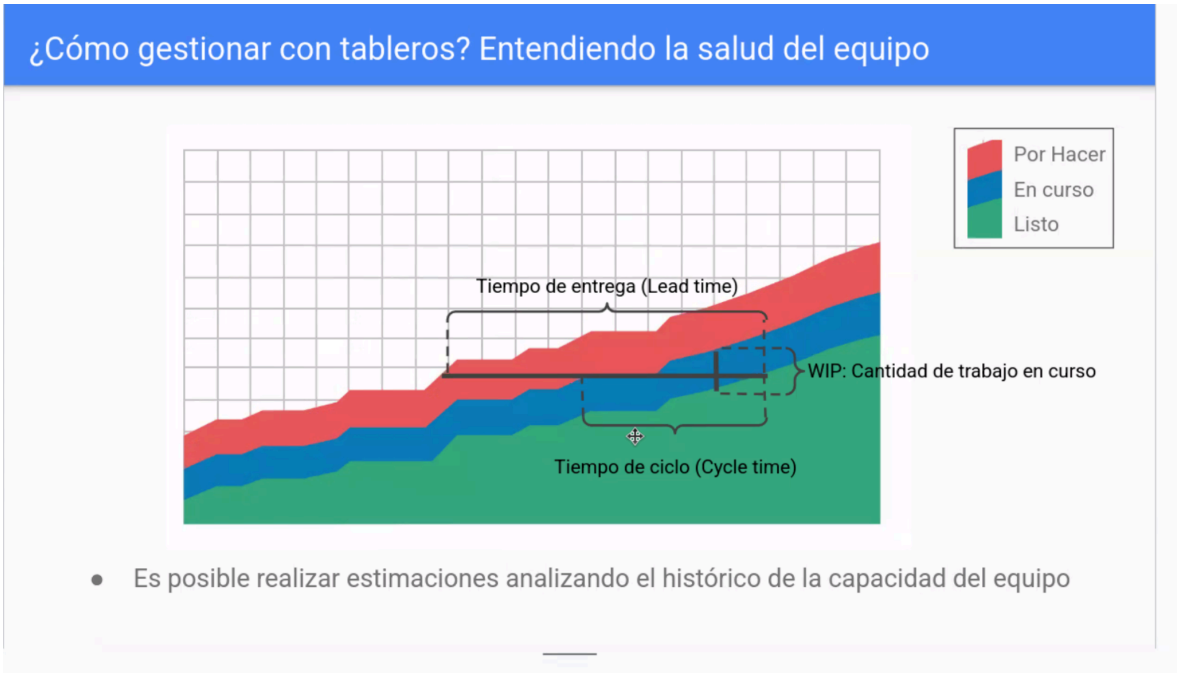


Tabla brindada por la Intendencia de Montevideo en la presentación de cómo miden la productividad del equipo.

Lead Time (Tiempo de Entrega)

El Lead Time se define como el período total transcurrido desde el momento en que una tarea es solicitada hasta que es entregada al cliente o usuario final, incluyendo todas las etapas intermedias como planificación, espera, ejecución y validación (Reinertsen, 2009). Este indicador es crucial para evaluar la capacidad de respuesta del equipo y su eficiencia para entregar valor de manera ágil y oportuna. Un Lead Time prolongado puede indicar la existencia de cuellos de botella, dependencias no gestionadas adecuadamente, o procesos ineficientes que impiden entregar resultados rápidamente (Anderson, 2010).

Cycle Time (Tiempo de Ciclo)

El Cycle Time corresponde al tiempo efectivo requerido para completar una tarea específica desde el momento en que el equipo comienza a trabajar en ella hasta su

finalización definitiva (Poppendieck & Poppendieck, 2003). Este indicador proporciona una medida directa de la eficiencia operativa del equipo, revelando posibles ineficiencias en la ejecución del trabajo como exceso de trabajo en curso, reuniones improductivas o una definición poco clara de criterios de finalización.

WIP (Trabajo en Progreso)

El Trabajo en Progreso (WIP, por sus siglas en inglés) se refiere a la cantidad de tareas que se encuentran simultáneamente activas o en ejecución en un momento dado. Un nivel elevado de WIP puede ser indicativo de sobrecarga en los equipos, problemas de priorización o prácticas inadecuadas de multitarea, lo que conduce generalmente a una disminución de la eficiencia y un retraso en la entrega de valor al cliente (Anderson, 2010).

Como se mencionó anteriormente, estas métricas se encuentran actualmente en fase piloto, aplicadas en algunos equipos que trabajan bajo metodologías ágiles, que aún representan una minoría dentro del área de desarrollo. En este contexto, la incorporación de inteligencia artificial no sólo permite potenciar el uso de estas métricas, sino también ampliar su aplicabilidad y valor como instrumentos de evaluación. Al integrar IA en los procesos, se facilita medir de manera más objetiva y sistemática el impacto de la automatización y la asistencia inteligente, contribuyendo a consolidar una cultura de mejora continua. De este modo, estas métricas no solo sirven para monitorear el desempeño actual, sino también para validar de forma empírica los beneficios derivados de la adopción de IA, apoyando la expansión progresiva de prácticas ágiles en la institución.

Problemática actual

Actualmente, el equipo enfrenta la problemática de contar con pocos recursos disponibles y diversidad de tecnologías, lo que impacta en la capacidad de respuesta y el desarrollo de

nuevas iniciativas. Ante esta situación, se evalúa la incorporación de inteligencia artificial como una alternativa para optimizar procesos, mejorar la eficiencia operativa y suplir las limitaciones de personal en distintas áreas.

5.2 Formas de usar la IA para mejorar la productividad

En el contexto actual del equipo de desarrollo de software de la Intendencia de Montevideo, caracterizado por una estructura consolidada, tecnologías diversas y un alto nivel de experiencia, la incorporación estratégica de herramientas basadas en inteligencia artificial (IA) puede incrementar significativamente tanto la eficiencia como la efectividad del equipo, contribuyendo directamente a una mejora en la productividad global. A continuación, se detallan las principales áreas donde la IA puede generar este impacto positivo.

Automatización y aceleración de tareas repetitivas

La automatización de tareas repetitivas constituye una de las áreas donde la inteligencia artificial puede generar beneficios inmediatos dentro del proceso de desarrollo de software. En los equipos de la Intendencia, una parte significativa del tiempo se destina a actividades repetitivas en diferentes contextos, como la generación de código estándar, la documentación técnica o la construcción de casos de prueba.

La incorporación de herramientas de IA permite optimizar estos procesos mediante la producción automática de componentes basados en patrones recurrentes, reduciendo la carga manual y mejorando la precisión. Esto libera a los desarrolladores para enfocarse en la lógica de negocio y en tareas más complejas, incrementando la productividad general.

Su aplicación resulta especialmente efectiva en proyectos con componentes repetitivos o de gran escala. Asimismo, en procesos de modernización tecnológica, como la migración

de sistemas construidos en Oracle Forms hacia entornos actuales, la IA puede asistir en la traducción y adaptación de código, reduciendo tiempos y riesgos.

En conjunto, la automatización mediante IA impacta positivamente en indicadores como el Cycle Time y el Lead Time, al acelerar las entregas, reducir errores manuales y promover la estandarización del código. Esto fortalece la calidad de los desarrollos, agiliza los procesos de revisión y consolida prácticas homogéneas dentro del equipo, posicionando a la institución hacia una gestión más eficiente e innovadora de sus proyectos de software.

Mejora en la calidad y precisión de los requerimientos

La etapa de levantamiento y definición de requerimientos es una de las fases más críticas en el ciclo de vida del desarrollo de software, dado que los errores, ambigüedades o inconsistencias que se produzcan en esta etapa inicial se traducen en retrabajos costosos y pérdida de eficiencia en fases posteriores. En el caso de la Intendencia de Montevideo, donde los proyectos de software involucran múltiples áreas funcionales, tecnologías heterogéneas y sistemas de misión crítica, la claridad en los requerimientos resulta indispensable para asegurar el éxito de las implementaciones.

La inteligencia artificial ofrece un aporte significativo en este ámbito mediante el uso de herramientas de procesamiento de lenguaje natural que permiten analizar documentos, especificaciones y solicitudes en lenguaje natural, identificando posibles ambigüedades o lagunas de información. Además, la IA puede asistir en la redacción de requerimientos más estructurados y claros, transformando insumos informales o dispersos en documentación técnica precisa y coherente. Esta capacidad no solo mejora la comunicación entre analistas funcionales y desarrolladores, sino que también garantiza que los equipos trabajen desde una base sólida, reduciendo los riesgos de malinterpretación y la necesidad de correcciones en etapas avanzadas.

La aplicación de este enfoque resulta especialmente efectiva en proyectos de gran impacto institucional, como aquellos relacionados con los sistemas SAP, donde la definición de procesos administrativos y financieros debe ser precisa para evitar inconsistencias en la gestión. También adquiere relevancia en los proyectos de desarrollo de portales institucionales, en los que la claridad de los requerimientos funcionales es fundamental para lograr una adecuada usabilidad y experiencia del ciudadano. En el ámbito de las aplicaciones móviles, la correcta especificación de casos de uso es igualmente crítica, ya que la falta de precisión en esta etapa puede repercutir en fallas de diseño o en funcionalidades poco alineadas con las necesidades de los usuarios.

En términos de impacto, la mejora en la calidad de los requerimientos contribuye a reducir los ciclos de retrabajo, optimiza el uso de los recursos humanos y técnicos, y aumenta la probabilidad de éxito de los proyectos. Además, al generar documentación más clara y completa desde el inicio, se facilita la trazabilidad de los requerimientos a lo largo de todo el ciclo de vida del software, lo que fortalece tanto la gestión del conocimiento como la capacidad de mantenimiento futuro de las aplicaciones.

La utilización de inteligencia artificial para mejorar la calidad y precisión de los requerimientos representa una oportunidad estratégica para la Intendencia de Montevideo, especialmente en proyectos de gran escala y alta criticidad. Su implementación no solo garantiza bases más sólidas para el desarrollo de software, sino que también contribuye a una gestión más eficiente y a la construcción de soluciones tecnológicas alineadas con las necesidades institucionales y de la ciudadanía.

Asistencia en el desarrollo de código

La etapa de codificación constituye uno de los pilares fundamentales del ciclo de vida del software, ya que es en esta instancia donde los requerimientos funcionales se transforman

en soluciones concretas. No obstante, este proceso suele ser intensivo en tiempo y propenso a errores, especialmente en un entorno complejo y heterogéneo como el de la Intendencia de Montevideo, donde conviven múltiples sistemas, arquitecturas y líneas de desarrollo. En este contexto, la incorporación de herramientas de inteligencia artificial orientadas a la asistencia en la programación representa una oportunidad estratégica para incrementar la eficiencia, reducir los tiempos de entrega y mejorar la calidad general de los productos desarrollados.

Los asistentes inteligentes de codificación, basados en técnicas de aprendizaje automático y en el reconocimiento de patrones de programación, pueden apoyar a los desarrolladores en diversas tareas: sugerir fragmentos de código, detectar errores en tiempo real, recomendar buenas prácticas y generar estructuras estandarizadas de manera automática. Este tipo de apoyo reduce la carga cognitiva del programador, favorece la homogeneidad del código y mejora su mantenibilidad, lo que se traduce en un desarrollo más ágil y confiable.

La aplicación de estas soluciones resulta especialmente valiosa en el entorno de la Intendencia, donde los equipos de desarrollo gestionan una amplia variedad de aplicaciones y servicios que deben integrarse entre sí para sostener los procesos institucionales. En estos casos, la asistencia inteligente puede acelerar la construcción de componentes y servicios, facilitar la configuración de entornos de despliegue y optimizar la gestión de datos, reduciendo el riesgo de errores humanos en etapas críticas del ciclo de desarrollo.

Desde una perspectiva organizacional, la adopción de inteligencia artificial en la etapa de codificación contribuye a la estandarización de los productos generados, disminuye la tasa de errores y potencia la calidad técnica del software. Además, promueve la transferencia de

conocimiento dentro de los equipos, permitiendo que los desarrolladores menos experimentados adquieran buenas prácticas a través de la interacción con las recomendaciones automatizadas.

En síntesis, la asistencia inteligente en el desarrollo de código constituye un recurso estratégico para los proyectos de software de la Intendencia de Montevideo. Su implementación impulsa la eficiencia operativa, mejora la calidad de los desarrollos y fortalece la capacidad institucional para responder a las crecientes demandas de modernización, digitalización y prestación de servicios tecnológicos confiables a la ciudadanía.

Optimización de procesos de prueba y validación

La etapa de pruebas y validación constituye un componente esencial en el ciclo de vida del software, dado que garantiza la calidad, estabilidad y confiabilidad de las soluciones implementadas. Sin embargo, en organizaciones con una infraestructura tecnológica amplia y diversa como la Intendencia, esta fase suele demandar grandes cantidades de tiempo y recursos humanos, en especial cuando se requiere cubrir escenarios complejos de uso y asegurar la interoperabilidad entre múltiples aplicaciones.

Las herramientas de prueba basadas en IA poseen la capacidad de generar casos de prueba automáticos a partir de los requerimientos o del código existente, reduciendo la dependencia de tareas manuales en el área de aseguramiento de la calidad. Además, son capaces de identificar patrones de error frecuentes, priorizar pruebas en función de los riesgos detectados y ajustar de manera dinámica los escenarios de validación conforme el software evoluciona. Esto favorece la detección temprana de fallos, minimiza la probabilidad de errores en producción y acorta los ciclos de retroalimentación entre los equipos de desarrollo y de calidad.

La aplicación de estas herramientas resulta especialmente efectiva en proyectos de gran criticidad institucional, como los sistemas SAP, donde la estabilidad de procesos financieros y administrativos requiere validaciones exhaustivas antes de cualquier actualización o despliegue. Del mismo modo, los desarrollos en Java, que involucran un gran número de microservicios interconectados, se beneficiarían de pruebas automatizadas continuas que permitan detectar rápidamente anomalías en la comunicación entre servicios. También en los portales institucionales y en las aplicaciones móviles , el uso de pruebas inteligentes permite asegurar la consistencia de la experiencia de usuario en diferentes dispositivos y escenarios de interacción.

El impacto de estas prácticas se refleja en una reducción significativa del Lead Time, al disminuir los tiempos de validación y facilitar un esquema de despliegue continuo (continuous delivery). Asimismo, el enfoque de pruebas inteligentes fortalece la confianza en el software entregado y reduce los costos asociados a correcciones tardías, que suelen ser las más complejas y costosas de abordar.

La optimización de los procesos de prueba y validación mediante IA representa un recurso estratégico para la Intendencia de Montevideo, particularmente en proyectos de alta criticidad y gran escala. Al automatizar la generación de pruebas, detectar fallos de forma temprana y priorizar los escenarios de validación más relevantes, la institución puede asegurar un software más robusto, confiable y alineado con las necesidades de los usuarios, consolidando a la vez una cultura de calidad y eficiencia en sus procesos de desarrollo tecnológico.

Mejora del monitoreo, operación y mantenimiento del software

Una vez que las aplicaciones entran en producción, el foco principal del ciclo de vida del software se desplaza hacia la operación y el mantenimiento. En esta etapa, el desafío

consiste en garantizar la continuidad del servicio, mantener niveles adecuados de rendimiento y prevenir incidentes que puedan afectar tanto a usuarios internos como a la ciudadanía. En un ecosistema complejo como el de la Intendencia de Montevideo, que administra más de ciento cincuenta aplicaciones basadas en Java y otros entornos tecnológicos heterogéneos, la gestión eficiente del monitoreo y la operación adquiere un rol estratégico.

La inteligencia artificial ofrece un aporte sustancial en este ámbito mediante herramientas de monitoreo inteligente, capaces de analizar grandes volúmenes de datos en tiempo real y detectar patrones anómalos antes de que se conviertan en incidentes críticos. Estas soluciones integran capacidades de análisis predictivo, correlación automática de eventos y generación de alertas proactivas, lo que permite a los equipos técnicos anticipar problemas y aplicar medidas correctivas antes de que los usuarios perciban una interrupción del servicio.

Su aplicación resulta especialmente efectiva en el monitoreo de las aplicaciones desplegadas en servidores, donde las variaciones en el consumo de recursos pueden generar inestabilidad si no son detectadas a tiempo. También es altamente relevante en la supervisión de contenedores y microservicios, donde la alta interdependencia entre servicios hace que un fallo aislado pueda propagarse rápidamente y afectar la disponibilidad general del sistema. En el caso de los sistemas SAP, de carácter crítico para la gestión administrativa, el uso de IA en la operación permitiría anticipar cuellos de botella en procesos de gran volumen de datos o detectar degradaciones en el rendimiento que podrían impactar en la ejecución de tareas esenciales.

El impacto de la implementación de estas herramientas se refleja en una reducción significativa de los tiempos de respuesta a incidentes y en la disminución del tiempo de

inactividad de los sistemas. Además, el enfoque predictivo favorece una transición hacia un modelo de mantenimiento proactivo, en el cual la atención no se centra únicamente en resolver problemas una vez ocurridos, sino en prevenir su aparición. Esto no solo mejora la experiencia de los usuarios internos y ciudadanos, sino que también contribuye a un uso más eficiente de los recursos humanos y tecnológicos de la institución.

Gestión eficiente del conocimiento y soporte automatizado

La gestión del conocimiento y el soporte a usuarios constituyen dimensiones fundamentales en organizaciones complejas como la Intendencia de Montevideo, donde coexisten múltiples sistemas, tecnologías y aplicaciones que requieren asistencia continua. Actualmente, una parte considerable de los recursos humanos del área de informática se destina a resolver incidencias recurrentes y consultas de usuarios internos, lo cual limita la capacidad del equipo para enfocarse en proyectos estratégicos. En este sentido, la incorporación de herramientas de inteligencia artificial orientadas al soporte y a la gestión del conocimiento ofrece un camino para optimizar estos procesos.

La implementación de chatbots y asistentes virtuales, basados en modelos de lenguaje y aprendizaje automático, permite responder de manera inmediata a consultas frecuentes, guiar a los usuarios en la resolución de problemas simples y derivar únicamente las incidencias más complejas al personal especializado. De este modo, se reduce la carga de trabajo del equipo de soporte, se mejora la eficiencia en la atención y se incrementa la satisfacción de los usuarios. A su vez, la IA facilita la construcción y actualización dinámica de bases de conocimiento, generando repositorios organizados que recopilan guías, buenas prácticas y documentación de uso.

Este tipo de soluciones resulta particularmente efectivo en proyectos relacionados con sistemas SAP, donde los usuarios finales suelen requerir apoyo para la ejecución de

transacciones, generación de reportes o resolución de problemas de acceso. También es relevante en el ámbito de los portales web institucionales y aplicaciones móviles, donde ciudadanos y funcionarios demandan asistencia inmediata en trámites o consultas frecuentes. Asimismo, la gestión de incidencias en plataformas de desarrollo y operación, como Jira Service Management, puede beneficiarse de la integración de asistentes inteligentes que resuelvan tickets de baja complejidad de manera automática.

En términos de impacto, la utilización de inteligencia artificial en la gestión del conocimiento y el soporte contribuye a reducir los tiempos de respuesta, aumentar la autonomía de los usuarios, y liberar al personal técnico para la atención de incidentes críticos o proyectos de innovación. Además, la estandarización de respuestas y la generación de documentación automatizada fortalecen la trazabilidad de la información, promueven el aprendizaje organizacional y favorecen la adopción de nuevas tecnologías dentro de la institución.

5.3 Riesgos y medidas de mitigación

La adopción de inteligencia artificial en la Intendencia de Montevideo presenta riesgos que abarcan desde la seguridad de la información hasta aspectos operativos y estratégicos. Entre los más relevantes se encuentran la posible filtración de datos sensibles, la introducción de vulnerabilidades en el código, la dependencia excesiva de las herramientas y los desafíos vinculados a la normativa y los derechos de autor. Para afrontarlos, resulta imprescindible definir políticas internas claras, fortalecer la capacitación del personal, mantener la supervisión humana obligatoria en los procesos críticos y establecer auditorías periódicas que aseguren un uso responsable, seguro y sostenible de estas tecnologías en el ámbito institucional.

5.3.1 Riesgos de seguridad de la información

Uno de los principales riesgos está vinculado a las fugas de información confidencial o sensible. En el caso de la IM, un analista funcional podría, de forma inadvertida, copiar y pegar fragmentos de código de SAP que contengan credenciales o tokens de acceso en una consulta a ChatGPT u otra herramienta de IA pública. Dado que estas plataformas funcionan en la nube, esa información quedaría expuesta fuera del entorno institucional. Para mitigar este riesgo es necesario establecer políticas internas claras que prohíban expresamente el ingreso de datos sensibles en estas herramientas, priorizar soluciones empresariales de IA que garanticen que la información no será usada para reentrenar modelos y aplicar prácticas de anonimización y cifrado antes de realizar cualquier prueba.

Otro riesgo está asociado a la exposición de datos personales y posibles incumplimientos normativos. El uso de herramientas IA en la nube podría implicar que se procesen datos de contribuyentes o información interna de la IM sin cumplir con lo establecido en la Ley N.º 18.331 de Protección de Datos Personales. Esto no solo supondría sanciones legales, sino también un impacto reputacional para la institución. La mitigación requiere controles de acceso estrictos, monitoreo del uso de IA y capacitación del personal en normativa de protección de datos, además de optar por entornos empresariales que permitan auditoría y trazabilidad de la información.

Asimismo, existe el riesgo de introducción de vulnerabilidades de seguridad en el código. Herramientas como Copilot pueden sugerir código funcional en lenguajes como PL/SQL o Java, pero con fallos de seguridad, como inyecciones SQL. En un entorno crítico como el de la IM, donde se mantienen más de 150 aplicaciones, una vulnerabilidad de este tipo podría afectar sistemas esenciales para la ciudadanía. La mitigación se basa en mantener

revisiones humanas obligatorias, incluir pruebas automatizadas de seguridad y utilizar listas de verificación alineadas con los estándares de OWASP.

Finalmente, se identifican los ataques a través de IA, en particular los de tipo prompt injection. Un caso plausible en la IM sería la manipulación de un bot de atención ciudadana impulsado por IA, donde un atacante logre introducir instrucciones ocultas para acceder a información no autorizada. Para evitarlo, se requiere un filtrado y sanitización de entradas, la incorporación de barreras de seguridad en los prompts y la realización de auditorías periódicas para detectar interacciones anómalas.

5.3.2 Riesgos operativos y estratégicos

En el plano operativo, un riesgo relevante es la generación de decisiones erróneas o sesgadas basadas en IA. Si el equipo de desarrollo utiliza pruebas automatizadas generadas por una IA entrenada con datos incompletos, es posible que se omitan casos críticos en aplicaciones de transporte o gestión tributaria, generando fallos en la prestación de servicios. Para mitigarlo, es fundamental que los equipos humanos validen los resultados, se capaciten en la detección de sesgos y que los modelos se ajusten con datasets locales que reflejen mejor la realidad de Montevideo.

También se identifica el riesgo de dependencia excesiva y pérdida de conocimiento interno. En un equipo con baja rotación como el de la IM, existe el peligro de que los desarrolladores deleguen en exceso la escritura de código a la IA, perdiendo expertise en plataformas críticas como JBoss, SAP o Lotus Domino. Esta dependencia podría dificultar la resolución de incidentes complejos sin apoyo externo. La mitigación pasa por fomentar la programación en pares, establecer políticas que equilibren el uso de IA con revisiones manuales, y reforzar la documentación y transmisión interna del conocimiento.

Otro punto crítico es la disminución del control de calidad en el desarrollo. El despliegue de aplicaciones en OpenShift con código generado automáticamente, sin mecanismos de validación adecuados, puede incorporar malas prácticas de programación y afectar la estabilidad de sistemas en producción. Para reducir este riesgo es necesario definir controles de calidad obligatorios en los procesos CI/CD, realizar auditorías de código y establecer estándares internos de legibilidad, seguridad y eficiencia que sirvan de referencia al momento de revisar lo generado por la IA.

Finalmente, se debe considerar el manejo del copyright y los derechos de autor. Herramientas como Copilot podrían generar fragmentos de código basados en repositorios GPL u otras licencias restrictivas, lo que colocaría a la IM en una situación legalmente riesgosa si ese código se incorpora sin atribución adecuada. La mitigación requiere registrar el origen de cada fragmento de código generado por IA, someterlo a revisión legal antes de ser incorporado en producción y capacitar al equipo en materia de licencias de software y propiedad intelectual.

5.3.3 Recomendaciones estratégicas para la IM

En el plano estratégico, se recomienda a la Intendencia de Montevideo adoptar un conjunto de medidas transversales que aseguren un uso responsable de la inteligencia artificial. En primer lugar, es imprescindible definir políticas internas claras sobre qué información puede o no ser utilizada en estas herramientas, bajo la supervisión de un comité de gobernanza de IA. Asimismo, se debe dar preferencia a soluciones empresariales que ofrecen mayores garantías de privacidad, trazabilidad y seguridad.

En paralelo, se propone implementar un plan de capacitación continua para los 90 integrantes del equipo, enfocado en el uso ético, legal y seguro de la IA, con especial atención a la normativa nacional vigente y a las limitaciones técnicas de estas

herramientas. La supervisión humana obligatoria debe ser un principio inamovible en el ciclo de desarrollo, particularmente en los sistemas críticos que utilizan SAP, Oracle, JBoss o microservicios desplegados en producción.

Por último, resulta indispensable que la IM incorpore auditorías periódicas sobre el uso de IA dentro de su sistema de gestión institucional, siguiendo estándares internacionales como ISO 27001 o el marco de gestión de riesgos del NIST. Estas auditorías deben no solo evaluar el cumplimiento de las políticas internas, sino también monitorear riesgos emergentes, adaptarse a cambios regulatorios y proponer mejoras continuas.

5.4 Plan incorporación de IA en los equipos de desarrollo

La implementación de inteligencia artificial debe llevarse adelante de forma progresiva, controlada y con base empírica. La estrategia más adecuada para avanzar en este proceso es la realización de un plan piloto, que permita validar la aplicabilidad de las herramientas, medir resultados y ajustar la estrategia antes de considerar su adopción generalizada. En este sentido, la Intendencia de Montevideo cuenta con una ventaja relevante: la existencia de un conjunto consolidado de métricas, entre ellas Lead Time, Cycle Time y WIP, que ya se utilizan para evaluar el desempeño de sus procesos de desarrollo. Estas métricas constituyen un marco ideal para medir el impacto concreto de la incorporación de IA.

5.4.1 Diagnóstico y Alcance del Piloto

El diagnóstico inicial debe comenzar con la identificación de un proyecto de pequeña escala, lo suficientemente representativo como para incluir tanto necesidades de documentación como la implementación de funcionalidades menores. Este tipo de proyectos permite validar la contribución de la IA en dos dimensiones fundamentales: la generación de documentación técnica actualizada para módulos ya existentes y la

asistencia en la implementación de nuevo código. La experiencia internacional muestra que este tipo de abordaje controlado es el más seguro y eficaz para evaluar herramientas emergentes, ya que permite minimizar riesgos y maximizar el aprendizaje institucional.

5.4.2 Selección y Configuración de Herramientas

Una vez definido el alcance del piloto, resulta necesario seleccionar las herramientas de IA más adecuadas para el contexto tecnológico de la Intendencia. En este caso, una combinación de soluciones empresariales, como ChatGPT Enterprise para la generación de documentación técnica y GitHub Copilot for Business para la asistencia en la codificación, representa una alternativa sólida. La configuración de estas herramientas debe realizarse en entornos controlados y sobre repositorios de prueba, lo que asegura la protección de datos institucionales y el cumplimiento de estándares de seguridad.

5.4.3 Capacitación y Sensibilización

Previo a la puesta en marcha, es fundamental un proceso de capacitación y sensibilización dirigido a los equipos involucrados. Dicho proceso debe incluir la validación crítica de los resultados generados por la IA, el fortalecimiento de buenas prácticas en materia de calidad de software y la incorporación de lineamientos éticos y legales relacionados con la propiedad intelectual y el uso de datos sensibles. De este modo, se promueve una cultura de adopción responsable que perciba a la IA como un apoyo complementario al conocimiento experto de los desarrolladores.

5.4.4 Ejecución del Piloto

La ejecución controlada del piloto se realizaría sobre un módulo reducido y de bajo riesgo, donde se apliquen las herramientas seleccionadas en tareas de documentación, generación

de casos de prueba básicos e implementación de mejoras menores en el código. Durante esta etapa se medirán indicadores que forman parte de la práctica habitual de la Intendencia. El Lead Time permitirá observar si el tiempo total desde la solicitud de un requerimiento hasta su entrega en producción se reduce gracias a la aceleración de la documentación inicial y la codificación de componentes repetitivos. El Cycle Time, por su parte, brindará información sobre la duración efectiva de las tareas desde el momento en que comienzan hasta que se completan, con la expectativa de que la asistencia de la IA contribuya a disminuir tiempos en codificación y pruebas. Finalmente, el WIP (Work in Progress) se utilizará para monitorear la cantidad de tareas en curso simultáneamente, con el objetivo de comprobar si el uso de IA contribuye a evitar cuellos de botella derivados de la acumulación de documentación y desarrollos menores pendientes.

5.4.5 Evaluación de Resultados

La evaluación de los resultados del piloto deberá realizarse comparando los valores obtenidos en estas métricas antes y después de la intervención, complementando el análisis con la percepción cualitativa de los equipos de desarrollo. Se espera que el Lead Time y el Cycle Time se reduzcan de manera significativa, mientras que el WIP muestre una disminución en la acumulación de tareas en curso. Asimismo, será relevante recoger retroalimentación sobre la facilidad de uso de las herramientas, el nivel de confianza en los resultados y el valor agregado percibido.

5.4.6 Recomendaciones y Escalado Progresivo

En caso de resultados positivos, la Intendencia podrá avanzar hacia un plan de escalado progresivo. Este deberá contemplar la definición de políticas institucionales para el uso de IA, auditorías periódicas de seguridad, privacidad y calidad de código, y la expansión

controlada hacia proyectos de mayor complejidad. El uso de métricas consolidadas como Lead Time, Cycle Time y WIP permitirá dar seguimiento riguroso a la efectividad del proceso, asegurando que la adopción de IA contribuya de manera tangible a la modernización tecnológica y a la mejora de la eficiencia operativa.

6 Conclusiones

La hipótesis planteada en este trabajo sostiene que la implementación de herramientas de inteligencia artificial en los equipos de desarrollo de software de la Intendencia de Montevideo incrementará su productividad, siempre que se gestione adecuadamente el cambio y se adopten medidas efectivas para proteger la seguridad de la información. Los resultados obtenidos permiten confirmar la validez de dicha hipótesis. El análisis realizado mostró que la utilización de herramientas de IA generativa y plataformas de automatización de pruebas y monitoreo tiene un efecto positivo en la reducción del Lead Time y del Cycle Time, en la optimización del WIP y en la mejora de la calidad de los productos de software entregados. No obstante, también se constató que estos beneficios solo pueden sostenerse en el tiempo si se acompañan de políticas institucionales claras, de la supervisión humana constante y de una estrategia de gestión del cambio que prepare al equipo para integrar estas tecnologías en forma ética, controlada y responsable.

En relación con el primer objetivo específico, referido a la identificación de las herramientas más pertinentes para el contexto institucional, la investigación concluyó que aquellas basadas en inteligencia artificial generativa son las que mejor se ajustan a las necesidades de la Intendencia. Soluciones como GitHub Copilot, ChatGPT Enterprise y asistentes especializados como Cursor y Devin ofrecen funcionalidades que impactan directamente en la automatización de código, en la generación de documentación y en la asistencia a tareas de soporte. En contraposición, los enfoques de aprendizaje automático tradicional, como la clasificación o el clustering, resultan menos adecuados para los desafíos que enfrenta el equipo de desarrollo de software.

En cuanto al segundo objetivo, orientado a evaluar la influencia de estas herramientas sobre la eficiencia y la efectividad del equipo, los hallazgos mostraron un impacto favorable tanto en la reducción de tiempos de programación, documentación y pruebas como en la mejora de la calidad y consistencia del código generado. Adicionalmente, la inteligencia artificial se reveló útil para fortalecer la gestión del conocimiento y agilizar el soporte a usuarios internos. Sin embargo, también se identificaron riesgos significativos, entre ellos la dependencia excesiva, la pérdida de conocimiento técnico acumulado y la introducción de vulnerabilidades de seguridad si no se mantiene una revisión humana adecuada.

El tercer objetivo, centrado en establecer métricas para medir el impacto real de la incorporación de la inteligencia artificial, permite confirmar que los indicadores ya utilizados por la Intendencia, Lead Time, Cycle Time y WIP, constituyen un marco sólido y aplicable. Complementar estas métricas con la percepción cualitativa de los equipos posibilita una evaluación más completa del valor generado por la adopción tecnológica.

Respecto al cuarto objetivo, destinado a determinar las fases del proceso de desarrollo más susceptibles de mejora mediante la inteligencia artificial, se identificó que la documentación y la definición de requerimientos se benefician al ganar en claridad y precisión, lo que reduce ambigüedades y retrabajos posteriores. La etapa de codificación se optimiza con la asistencia de herramientas que aceleran la escritura de código y promueven la estandarización de prácticas. Las fases de pruebas y validación también resultan favorecidas gracias a la generación automatizada de casos de prueba y a la detección temprana de errores, mientras que el monitoreo y el soporte se fortalecen mediante la implementación de chatbots y sistemas predictivos de mantenimiento.

Finalmente, en relación con el objetivo orientado al diseño de estrategias de cambio que aseguren una transición exitosa y sostenible, la investigación concluye que la adopción de inteligencia artificial no puede limitarse a la dimensión técnica, sino que requiere una estrategia integral de gestión del cambio. Dicha estrategia debe contemplar la comunicación clara de objetivos y beneficios, la capacitación continua del equipo en el uso seguro, legal y ético de las herramientas, la aplicación de pilotos controlados como etapa inicial de implementación, la obligatoriedad de la supervisión humana en los procesos críticos y la realización de auditorías periódicas que garanticen calidad, seguridad y cumplimiento normativo. Con estas medidas, la Intendencia no solo reduce resistencias internas y asegura la aceptación cultural de la innovación, sino que también establece las bases para una integración sostenible de la inteligencia artificial en sus procesos de desarrollo de software.

7 Trabajos a Futuro

A partir de los hallazgos de esta investigación, resulta pertinente plantear diversas líneas de trabajo a futuro que permitan consolidar y ampliar los avances alcanzados en la incorporación de inteligencia artificial en los equipos de desarrollo de software de la Intendencia de Montevideo. Una primera línea consiste en la realización de estudios longitudinales que permitan evaluar el impacto sostenido de estas herramientas sobre las métricas de productividad institucional. En particular, se vuelve necesario observar cómo la utilización de inteligencia artificial afecta en el mediano y largo plazo indicadores como el Lead Time, el Cycle Time, la calidad del código y la satisfacción del equipo de desarrollo. Este seguimiento prolongado permitirá determinar si los beneficios iniciales se mantienen y en qué medida surgen nuevas problemáticas asociadas al aprendizaje, la dependencia tecnológica o la pérdida de conocimiento interno.

Una segunda línea de trabajo se orienta a la comparación sistemática de diferentes herramientas disponibles en el mercado. Analizar de forma empírica el desempeño de soluciones como GitHub Copilot, Amazon CodeWhisperer, TabNine o nuevas plataformas emergentes permitiría construir criterios más objetivos para la selección de tecnologías según el lenguaje de programación utilizado, el tipo de proyecto en curso y el nivel de experiencia de los desarrolladores. Este tipo de análisis comparativo sería particularmente valioso en un entorno heterogéneo como el de la Intendencia, donde coexisten tecnologías diversas que abarcan desde Java y SAP hasta entornos de microservicios en OpenShift.

Un tercer ámbito de investigación futura corresponde al desarrollo de marcos normativos y guías éticas específicas para el uso de inteligencia artificial en instituciones públicas. La definición de lineamientos claros sobre privacidad de datos, seguridad del código

generado, transparencia algorítmica y responsabilidad institucional permitiría dar mayor solidez y legitimidad a los procesos de adopción tecnológica. Estos marcos regulatorios deberían ajustarse tanto a la legislación nacional vigente como a las normativas internacionales, considerando las particularidades de los organismos estatales en el manejo de información crítica de la ciudadanía.

Asimismo, resulta de interés explorar la escalabilidad del modelo de adopción de inteligencia artificial hacia otras áreas de la Intendencia o incluso hacia organismos públicos con problemáticas similares. Evaluar la replicabilidad de los aprendizajes obtenidos en distintos contextos permitirá validar la utilidad del enfoque, identificar limitaciones y proponer ajustes que favorezcan una transformación digital más amplia y consistente en el sector público.

En conjunto, estos trabajos apuntan a fortalecer una adopción de inteligencia artificial que sea sostenible, segura, medible y alineada con los principios de responsabilidad institucional, innovación tecnológica y mejora continua de los servicios públicos.

8 Referencias bibliográficas y bibliografía

A03 injection. (n.d.). OWASP Top 10:2021. Retrieved April 20, 2025, from https://owasp.org/Top10/A03_2021-Injection/

Accenture. (n.d.). With 12,000 developers using GitHub Copilot, Accenture doubles down on GitHub's platform. *GitHub*. Retrieved June 2, 2025, from https://github.com/customer-stories/accenture

AI risk management framework. (2021, July 12). NIST. https://www.nist.gov/itl/ai-risk-management-framework

AI risk management framework. (2023, January 26). NIST. https://www.nist.gov/itl/ai-risk-management-framework

Anderson, D., & Anderson, L. A. (2010). *Beyond Change Management: How to achieve breakthrough results through conscious change leadership*. John Wiley & Sons.

Anderson, D. J. (2010). *Kanban: Cambio evolutivo exitoso para su negocio de tecnología*.

Ashare, M. (2025, January 16). Citi deploys AI coding tools to 30K developers in modernization push. *CIO Dive*. https://www.ciodive.com/news/citi-bank-modernization-generative-ai-coding-assistant/737624/

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., …

Liang, P. (2021, August 16). *On the opportunities and risks of foundation models*. arXiv.Org. https://arxiv.org/abs/2108.07258

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020, May 28). *Language Models are Few-Shot Learners*. arXiv.Org. https://arxiv.org/abs/2005.14165

Ebert, C., & Louridas, P. (2023). Generative AI for software practitioners. *IEEE Software*, *40*(4), 30–38. https://doi.org/10.1109/ms.2023.3265877

Frank, E., & Godwin, O. (2024). *Enhancing Developer Productivity: a Study on GitHub Copilot's Code Completion Capabilities*.

Fu, Y., & Soman, C. (2021, March 31). *Real-time data infrastructure at Uber*. arXiv.Org. https://arxiv.org/abs/2104.00087

General Data Protection Regulation (GDPR) – legal text. (2016, July 13). General Data Protection Regulation (GDPR). https://gdpr-info.eu/

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014, June 10). *Generative adversarial networks*. arXiv.Org. https://arxiv.org/abs/1406.2661

Hayes, J. (2021). *The theory and practice of change management*. Bloomsbury Publishing.

Hiatt, J. (2006). *ADKAR: A model for change in business, government, and our community*.

IEEE standard for software quality assurance processes. (n.d.). https://doi.org/10.1109/ieeestd.2014.6835311

Kotter, J. P. (2012). *Leading change*. Harvard Business Press.

Kotter, J. P., & Cohen, D. S. (2012). *The Heart of Change: Real-Life stories of how people change their organizations*. Harvard Business Press.

Kuhail, M. A., Mathew, S. S., Khalil, A., Berengueres, J., & Shah, S. J. H. (2024). "Will I be replaced?" Assessing ChatGPT's effect on software development and programmer perceptions of AI tools. *Science of Computer Programming*, *235*, 103111. https://doi.org/10.1016/j.scico.2024.103111

Ley N° 18331. (n.d.). Retrieved April 20, 2025, from https://www.impo.com.uy/bases/leyes/18331-2008

Ma, Q., Wu, T., & Koedinger, K. (2023, June 8). *Is AI the better programming partner? Human-Human Pair Programming vs. Human-AI pAIr Programming*. arXiv.Org. https://arxiv.org/abs/2306.05153

March 2023. (n.d.). Microsoft Security Blog. Retrieved April 20, 2025, from https://www.microsoft.com/en-us/security/blog/2023/03/

Mbizo, T., Oosterwyk, G., Tsibolane, P., & Kautondokwa, P. (2024). Cautious optimism: The influence of generative AI tools in software development projects. In *Communications in Computer and Information Science* (pp. 361–373). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-64881-6_21

NguyenDuc, A., Cabrero-Daniel, B., Arora, C., Przybylek, A., Khanna, D., Herda, T., Rafiq, U., Melegati, J., Guerra, E., Kemell, K.-K., Saari, M., Zhang, Z., Le, H.,

Quan, T., & Abrahamsson, P. (2023). *Generative artificial intelligence for software engineering - A research agenda*. Elsevier BV. https://doi.org/10.2139/ssrn.4622517

Norvig, P., & Russell, S. (2021). *Artificial Intelligence: A Modern Approach, Global Edition*.

OWASP AI security and privacy guide. (n.d.). OWASP Foundation. Retrieved April 20, 2025, from https://owasp.org/www-project-ai-security-and-privacy-guide/

Page, K. (2015). *Making Sense of Change Management: A Complete Guide to the Models, Tools and Techniques of Organizational Change* (4th ed.). (2015)

Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2021, August 20). *Asleep at the keyboard? Assessing the security of GitHub copilot's code contributions*. arXiv.Org. https://arxiv.org/abs/2108.09293

Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2022). Asleep at the keyboard? Assessing the security of GitHub copilot's code contributions. *2022 IEEE Symposium on Security and Privacy (SP)*, 754–768. https://doi.org/10.1109/sp46214.2022.9833571

Poppendieck, M., Poppendieck, T. D., & Poppendieck, T. (2003). *Lean software development: An agile toolkit*. Addison-Wesley Professional.

Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach*.

Reinertsen, D. G. (2009). *The principles of product development flow: Second generation lean product development*. Celeritas Pub.

Responsible AI. (n.d.). SAP. Retrieved June 2, 2025, from https://www.sap.com/products/artificial-intelligence/ai-ethics.html

Sommerville, I. (2016). *Software engineering, global edition*. Pearson Higher Ed.

Taylor, A. (2025, April 22). *How real-world businesses are transforming with AI — with 261 new stories*. The Official Microsoft Blog. https://blogs.microsoft.com/blog/2025/04/22/https-blogs-microsoft-com-blog-2024-11-12-how-real-world-businesses-are-transforming-with-ai/

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017, June 12). *Attention is all you need*. arXiv.Org. https://arxiv.org/abs/1706.03762

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., & Irving, G. (2019, September 18). *Fine-Tuning language models from human preferences*. arXiv.Org. https://arxiv.org/abs/1909.08593