

Productividad en Frameworks Web para el desarrollo de dashboards interactivos con Power BI Pro

Productivity in Web Frameworks for developing interactive dashboards with Power BI Pro

Gladys Geoconda Esquivel Paula¹[0009-0002-3715-777], Luis René Quisaguano Collaguazo²[0000-0003-1345-0898], Abraham Paúl Caluña Guamán³[0009-0004-4065-0283], Víctor Alfonso Chiguano Lagla⁴[0009-0005-9086-5829]

¹ Universidad Técnica de Cotopaxi – Ecuador, gladys.esquivel9@utc.edu.ec.

² Universidad Técnica de Cotopaxi – Ecuador, abraham. luis.quisaguano1@utc.edu.ec.

³ Universidad Técnica de Cotopaxi – Ciencias de la Ingeniería y Aplicadas – Ingeniería en Sistemas de la Información– Ecuador, abraham.caluna8057@utc.edu.ec.

⁴ Universidad Técnica de Cotopaxi – Ciencias de la Ingeniería y Aplicadas – Ingeniería en Sistemas de la Información– Ecuador, victor.chiguano7102@utc.edu.ec.

CITA EN APA:

Esquivel Paula, G. G., Quisaguano Collaguazo, L. R., Caluña Guamán, A. P., & Chiguano Lagla, V. A. Productividad en Frameworks Web para el desarrollo de dashboards interactivos con Power BI Pro. Technology Rain Journal, 4(2). <https://doi.org/10.55204/trj.v4i2.e78>

Recibido: 12 de mayo-2025

Aceptado: 18 de julio-2025

Publicado: 23 de julio-2025

Technology Rain Journal

ISSN: 2953-464X

Resumen.

Introducción: El uso de dashboards es muy popular en la actualidad por las empresas, lo interesante es determinar cual es el tipo de tecnología base idóneo para los diferentes proyectos. **Objetivo:** analizar la eficiencia técnica y operativa de dos sistemas generadores de dashboards interactivos con diferente estructura determinando su desempeño desarrollo y capacidad de integración con Power BI Pro. **Metodología:** Investigación aplicada y comparativa. Técnicas empleadas Observación directa, medición de tiempos, pruebas de funcionalidad. Las dimensiones de estudio fueron desarrollo, productividad – escalabilidad y desempeño – rendimiento. **Resultados:** Se desarrollaron dos sistemas uno mediante .Net alojado en Windows y otro con Django alojado en Linux los dos fueron conectados con Power BI Pro para la generación de Dashboards interactivos de un entorno de empresa simulado. los dos sistemas generan entornos visuales semejantes, las diferencias sustanciales radican en los tiempos de desarrollo que son mayores en el sistema Django al estar alojado en Linux requirió una estructura adicional para integrarse a Power Bi Pro, en desempeño del sistema mediante la medición del tiempo de carga de .Net fue de 3.2 segundos a diferencia de Django 46 segundos, .net presentó un error técnico y Django 3. **Conclusión:** La elección sobre la mejor opción depende del contexto, la disponibilidad de recursos y los objetivos que se plantee para el proyecto .Net es eficiente en entornos donde se requiere mayor rapidez y compatibilidad nativa con Microsoft mientras que Django es útil en proyectos que libres que se enfoquen en la personalización.

Palabras Clave: Productividad, Frameworks Web, Dashboards, Power BI Pro.

Abstract:: Introduction: The use of dashboards is very popular nowadays among companies, the interesting thing is to determine which is the ideal type of base technology for different projects. Objective: To analyze the technical and operational efficiency of two interactive dashboard generating systems with different structures, determining their development performance and integration capacity with Power BI Pro. Methodology: Applied and comparative research. Techniques used: Direct observation, time measurement, functionality testing. The study dimensions were development, productivity - scalability and performance - performance. Results: Two systems were developed, one



Los contenidos de este artículo están bajo una licencia de Creative Commons Attribution 4.0 International (CC BY 4.0)

Los autores conservan los derechos morales y patrimoniales de sus obras.

using .Net hosted on Windows and the other with Django hosted on Linux, both were connected to Power BI Pro to generate interactive Dashboards of a simulated business environment. the two systems generate similar visual environments, the substantial differences lie in the development times that are longer in the Django system as being hosted on Linux required an additional structure to integrate with Power BI Pro, in system performance by measuring the loading time of .Net was 3.2 seconds unlike Django 46 seconds, .Net presented a technical error and Django 3. Conclusion: The choice on the best option depends on the context, the availability of resources and the objectives that are set for the project. .Net is efficient in environments where greater speed and native compatibility with Microsoft are required while Django is useful in free projects that focus on customization.

Keywords: Productivity, Web Frameworks, Dashboards, Power BI Pro.

1. INTRODUCCIÓN

“La rápida innovación tecnológica ha revolucionado todos los campos de la sociedad actual” (Hidalgo Achig et al., 2021) en el contexto actual de transformación digital la visualización de datos se posiciona como una herramienta eficiente para la toma de decisiones estratégicas empresariales, al ofrecer un entorno corporativo más eficiente y ágil con datos es Power BI Pro, por su funcionalidad presenta los datos de forma interactiva y llamativa ofrece indicadores y métricas claves refleja información relevante óptima para la gestión de empresas de cualquier tamaño y naturaleza económica.

Sin embargo, es importante considerar que la funcionalidad de esta herramienta interactiva depende significativamente del framework web que se utilice en el desarrollo, y de la complementariedad de los sistemas de backend que conformen la lógica del negocio al que se enfoca la herramienta.

En este contexto la dificultad se presenta en la elección del framework, especialmente cuando el entorno del negocio es complejo y requieren controlar diversos elementos como inventarios, ventas, proveedores, clientes, etc. es necesario que se obtengan métricas claras y precisas de la inter operatividad y rendimiento del negocio, la decisión sobre que framework se empelará debe basarse en datos reales del funcionamiento y operatividad para evitar costes innecesarios, mayores tiempos de desarrollo y restricciones futuras en el uso de la herramienta.

En el área existen diversos estudios que se enfocan en analizar de forma aislada o individual el uso del uso de .NET o Django con la integración de Power BI Pro para la generación de reportes empresariales dinámicos. No obstante, son escasos aquellos estudios que vinculan el desarrollo de dashboards integrados y su eficiencia con la elección del framework. Por lo expuesto, la investigación se enfoca en desarrollar dos estructuras paralelas la una con .NET/SQL Server/Windows y otra con Django/PostgreSQL/Linux en ambos casos conectadas a Power BI Pro para evaluar su eficiencia en la generación de métricas empresariales bajo elementos como la

experiencia del usuario final, el tiempo de desarrollo, la facilidad de integración, escalabilidad, interfaz, etc.

El estudio tiene como objetivo analizar la eficiencia técnica y operativa de dos sistemas generadores de dashboards interactivos con diferente estructura determinando su desempeño desarrollo y capacidad de integración con Power BI Pro. Además, se pretende responder las siguientes interrogantes científicas ¿Cuál de las dos estructuras genera una integración más rápida y eficiente con Power BI Pro? ¿Existen variaciones en el desarrollo de respuesta y rendimiento en la generación de dashboards interactivos? ¿Existen diferencias significativas en la estructura de un sistema alojado en Windows y otro bajo Linux?

Y comprobar la hipótesis: Un entorno basado en Python Framework Django genera la misma eficiencia y escalabilidad que un entorno desarrollado en Framework Web Forms para el desarrollo de dashboards interactivos mediante integración a Power BI Pro.

Para efecto se empleará un entorno de venta simulado, de esta manera se compara las dos estructuras para esclarecer la eficiencia en la generación de indicadores y métricas interactivas estableciendo ventajas y desventajas técnicas y operativas de los dos sistemas, esta información es especialmente útil para la toma de decisiones. Considerando lo que afirma (Quishpe Zapata, 2024) en la actualidad las tecnologías, actividades, herramientas que ofrecen visualización, informes, análisis y transformación de datos para la correcta toma de decisiones empresariales económicas, estratégicas de nivel operacional o gerencial es una necesidad imperiosa para la sostenibilidad empresarial. Como punto inicial se analiza ASP.NET (.NET FRAMEWORK) Framework Web Forms y Python Framework Django. En segunda instancia se presentan las bases de datos SQL Server y Postores SQL y finalmente Power BI Pro.

1.1 ASP.NET (.Net Framework) y Python Framework Django

Al analizar la evolución que ha tenido .NET como afirman (Narváez Vilema et al., 2025) su historia inicia en el año 2002 cuando Microsoft lanzó su primera versión, que se enfocaba a integrar varios lenguajes y herramientas para hacer más fácil el desarrollo de aplicaciones en Windows, para habilitar la programación en múltiples lenguajes se incluyó el Common Language Runtime (CLR) En el año 2016 se lanzó .NET Core que consistió en una versión multiplataforma esto permitió la integración en Linux, Windows, etc. Esto cambió de forma marcada la filosofía de programación orientándose al código abierto. Y, en 2020 se integraron las dos versiones ofreciendo la posibilidad de programar en la web, en la nube y en dispositivos móviles desde una misma aplicación.

Es importante considerar lo que manifiesta (Lombana Perez, 2021) en el desarrollo se debe relacionar el frontend y el backend a fin de obtener herramientas netamente funcionales que ofrezcan una experiencia eficiente e interactiva, accesible y atractiva. El primero hace referencia a los elementos que interactúan con el usuario de forma directa mientras que el segundo elemento es la

lógica y la gestión de datos se comunica con el frontend y ejecuta las acciones que el usuario requiere y debe ser capaz de procesar diversas solicitudes de forma simultánea y poseer un nivel de seguridad aceptable.

Por su parte Python de acuerdo a (IT Madrid, 2021) es un lenguaje de programación con semántica dinámica y de alto nivel, favorece la producción por su nivel de legibilidad de código y su carácter Open Source, lo sitúan como el líder sobre todo en el Data Science porque posee variedad de bibliotecas.

1.2 SQL Server y Postgres SQL

SQL es un sistema que según (Angulo Pinedo, 2021) integra tablas para ordenar y almacenar diversos datos y estructurarlos de formas específicas. En las tablas se encuentran filas y columnas que se conocen también como o tuplas o registros y atributos respectivamente. Cada una de las columnas se establecen con criterios específicos de datos como: cantidades monetarias, números, fechas, nombres, etc. En este sistema se puede encontrar MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database y SQLite y pueden ser empleadas de acuerdo con las funcionalidades y requerimientos desde cuestiones empresariales hasta páginas web.

SQL Server emplea el lenguaje Transact SQL lo que según (Morales Sevilla, 2021) permite manejar datos mediante lógica procedural, es un sistema que ofrece un nivel de seguridad eficiente y robusto demostrando escalabilidad y alta disponibilidad. Además, su interfaz gráfica favorece la ejecución y administración de comandos SQL, en este contexto se pueden desarrollar tareas complejas sin tener que sobre escribir el código.

SQL dispone de distintas ediciones que según (Abhishek Parandkar, 2024) lo sitúan como ideal para las empresas de grande tamaño mediante Enterprise y para aquellas pequeñas o medianas la edición Estándar y sin olvidar la opción gratuita que permite experimentar todas sus funcionalidades sin incurrir en costo alguno.

1.3 Power BI Pro

Es una herramienta creada por Microsoft enfocada en ofrecer visualización de datos de forma interactiva y puede ser empleada desde dispositivos móviles. Es el elemento que hace posible la interactividad en los dashboards, su interfaz es amigable y sencilla, impulsa el trabajo en equipo porque permite compartir informes y trabajar en tiempo real (Briones Acuña et al., 2024). Está compuesta de tres elementos principales:

Figura 1

Elementos de Power BI Pro



Nota: Elaboración propia a partir de Narvaes et al., 2025.

Power BI Pro es una herramienta posicionada en la gestión, administración y visualización de datos, por su alcance puede manejar cualquier tipo de datos de diferentes orígenes y de cualquier complejidad. Los componentes, diseños y recursos atractivos que posee en la versión actual hacen más fácil e interactiva la comprensión e interpretación de datos (Pacheco Huanca & Dionicio Asto, 2024). Power BI Pro de acuerdo a (Álvarez Medina, 2022, p.12) es una herramienta líder en inteligencia de negocios que se destaca por su integración con otras aplicaciones de Microsoft y su capacidad para generar informes interactivos y dashboards personalizables.

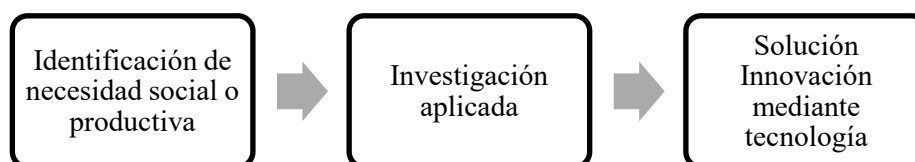
La conexión de Power BI Pro y SQL Server según (Álvarez Medina, 2022) se efectúa mediante una integración sólida y estable entre el programa y la base de datos esto hace posible un flujo continuo de información sincronizando los datos en tiempo real, SQL Server es el almacén donde se reserva gran cantidad de datos y Power BI Pro es la ventana por la que se refleja mediante gráficos e informes interactivos la información haciéndola más visual y atractiva.

2. METODOLOGÍA O MATERIALES Y METODOS

La investigación se plantea como aplicada porque se enfoca en “un objetivo práctico y específico, en un contexto determinado con problemas de la vida cotidiana y práctica, aplicando o utilizando el conocimiento de una o varias áreas especializadas para dar solución a un problema social o productivo”. En este contexto se enfoca en brindar una solución específica e innovadora para la toma de decisiones empresariales mediante el desarrollo de dashboards interactivos con Power BI Pro enfocado a un entorno comercial específico simulado. A continuación, se presenta la dinámica de la investigación aplicada:

Figura 2

Dinámica de la investigación aplicada



Además, es comparativa porque se desarrollan dos tipos de estructura con diferente lenguaje de programación y base de datos para determinar el rendimiento, eficiencia y comportamiento de cada framework. Como técnicas de investigación se aplicó:

- La observación directa y estructurada se enfoca en recolectar datos sobre la organización de códigos y elementos visibles del proceso de desarrollo
- Evaluación y medición de la productividad y rendimiento – tiempos de respuesta.

- Pruebas de funcionalidad de los dos sistemas con los mismos elementos para contrastar la información y establecer variaciones sobre la ejecución en condiciones reales.
- Técnica estadística t-student que permite medir la variación entre los dos sistemas y aceptar o rechazar la hipótesis de estudio.

Las dimensiones e indicadores empleados para la comparativa serán:

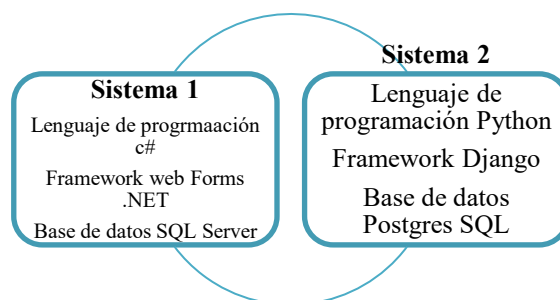
Tabla 1

Dimensiones e indicadores analizados

Dimensión	Elementos Evaluados	Indicadores Clave	Técnicas / Instrumentos	Escala de Medición
Desarrollo	Framework utilizado Tiempo de desarrollo	Tipo de framework Tiempo total de codificación	Observación directa (ficha técnica) Evaluación de desempeño (hoja de control de tiempo)	Nominal - Razón
Productividad y Escalabilidad	Facilidad de integración con Power BI Modularidad del código	Tiempo de integración con Power BI Separación lógica de componentes	Pruebas de funcionalidad (registro de integración) Observación estructurada (lista de verificación)	Razón - Ordinal
Desempeño del sistema (Rendimiento)	Tiempo de carga Estabilidad del sistema Tiempo de respuesta del servidor	Tiempo de carga del dashboards Número de errores Tiempo de ejecución (API/consulta)	Pruebas de funcionalidad (cronómetro/F12 navegador) Observación (registro de incidencias) Evaluación (Postman, consola navegador)	Razón

Se eligieron los indicadores porque permiten realizar una evaluación integral de elementos claves del funcionamiento, desarrollo y rendimiento de los sistemas, analizando la tecnología base que se empleó, el tiempo que tomó el desarrollo y la eficiencia de integración con la herramienta analítica Power BI Pro, los tiempos de carga en consultas y respuestas y la estabilidad que presenten y ofrecen información que es claramente comparable y contrastable para definir la alternativa más eficiente desde diversas perspectivas.

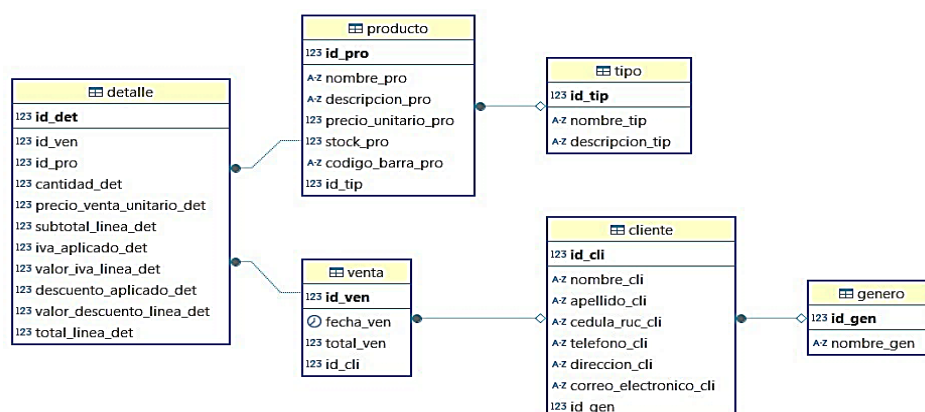
Se emplearon dos tipos de framework para cada sistema como se detalla a continuación:

Figura 3*Elementos de cada sistema*

El primer sistema .NET fue desarrollado y alojado en Windows, mientras que el segundo Django se desarrolló y alojó en Linux/Fedora. Los dos sistemas se conectaron a Power BI Pro un software nativo de Windows.

El modelo de datos o diagrama entidad relación (ERD) que se empleó en los dos sistemas establece conexión clara entre tablas contiene lo siguiente:

1. En detalle se almacena elementos de cada venta como los precios, cantidad, productos, IVA.
2. Tipo agrupa los productos según la categoría a la que pertenece.
3. Producto guarda información que caracteriza y diferencia cada producto como el precio, stock, nombre, etc.
4. Venta asocia los datos de las transacciones la fecha en que se realizó el cliente, el valor, etc.
5. Cliente mantiene registros de los datos personales del cliente.
6. Género clasifica a los clientes y agrupa de acuerdo con su género.

Figura 4*Diagrama empresa - relación*

3. RESULTADOS Y DISCUSIÓN

En este apartado se exponen los resultados en el desarrollo de los dos sistemas y posterior se presentan los indicadores que se obtienen de la integración con Power BI Pro. Para finalmente exponer los indicadores analizados en contraste y determinar la eficiencia de los dos sistemas.

3.1 Ficha técnica de los dos sistemas

Tabla 2

Ficha técnica de los sistemas

Ficha técnica de los sistemas		
	Sistema 1 Windows	Sistema 2 Linux
Framework	.NET Framework Web Forms	Django MVC
Lenguaje de programación	C#	Python
Base de datos	SQL Server	PostgreSQL
Sistema Operativo	Windows	Linux
Integración con Power BI Pro	Directa, tecnologías nativas	Entorno mixto de trabajo

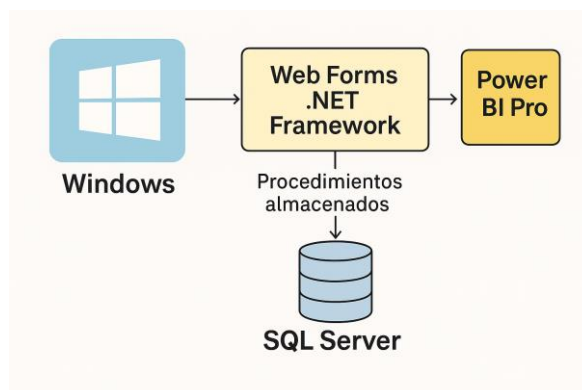
3.2 Sistema 1 .Net

Alojamiento del sistema 1: Para el sistema 1 se utilizó Windows como medio de alojamiento aprovechando su compatibilidad con tecnologías desarrolladas por Microsoft.

Arquitectura empleada: En este sistema se empleó Web Forms bajo el .NET Framework que consiste en el uso de formularios para generar interfaces reutilizables.

Base de datos: Se empleó como base de datos **SQL Server** mediante procedimientos almacenados para las acciones CRUD: Eliminar, Guardar, editar. Esto contribuye a robustecer la seguridad del sistema.

Conexión y comunicación con Power BI Pro: Como el sistema se alojó en Windows y Power BI es nativo de Windows la conexión no requirió adaptaciones ni capas intermedias.

Figura 5*Descripción gráfica del sistema 1*

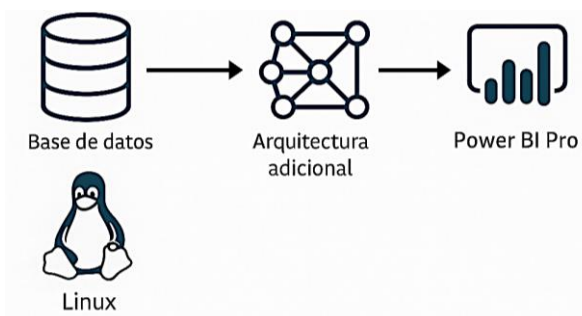
3.3 Sistema 2 Django

Alojamiento del sistema 2: El segundo sistema se alojó en Linux que es un sistema de código abierto.

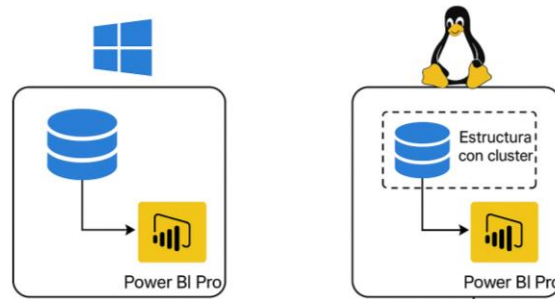
Arquitectura: Se empleó el Framework Django con arquitectura MVC (Modelo-Vista-Controlador) en lenguaje Python. De forma general compone una estructura limpia con elementos establecidos de presentación de datos y lógica.

Base de datos: Se utilizó PostgreSQL que es considerado una base de datos avanzada y con un elevado nivel de compatibilidad con Django lo que consolida como una opción eficiente para generar sistemas robustos.

Conexión con Power BI Pro: Debido a que Power BI Pro no se basa en código abierto, para lograr la comunicación entre estas dos tecnologías fue necesario crear una pequeña estructura que permitió abrir puertos mediante clúster esto hizo posible la comunicación cruzada mediante un entorno mixto de trabajo.

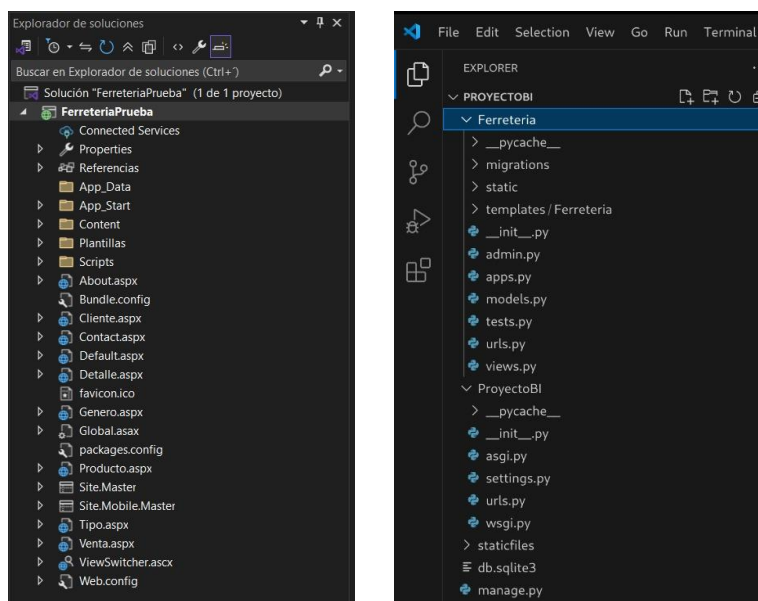
Figura 6*Descripción gráfica del sistema 2*

En resumen, la diferencia en la conexión de los dos sistemas con Power BI Pro fue:

Figura 7*Diferencia entre el sistema 1 y 2*

3.4 Estructuras empleadas en los dos sistemas

A la izquierda se refleja la estructura de carpetas de .Net Windows y a la derecha arquitectura MVC para Django. En la arquitectura .Net los archivos .aspx representan las páginas web vinculadas mediante codificación backend en C#, cuenta también con carpetas como App_Data, Scripts, Content y Plantillas, Web.config que es una carpeta de configuración lo que se configura como estructuras tradicionales con un enfoque visual. La segunda imagen refleja una configuración basada en Modelo-Vista-Controlador (MVC) con carpetas elementales del framework models.py, views.py, urls.py y admin.py que enmarcan la lógica del sistema además de otras carpetas características de Django migrations, static, y templates, esta imagen refleja una arquitectura enfocada en la separación de funciones de formaq modular lo que incrementa la escalabilidad y mantenimiento del proyecto.

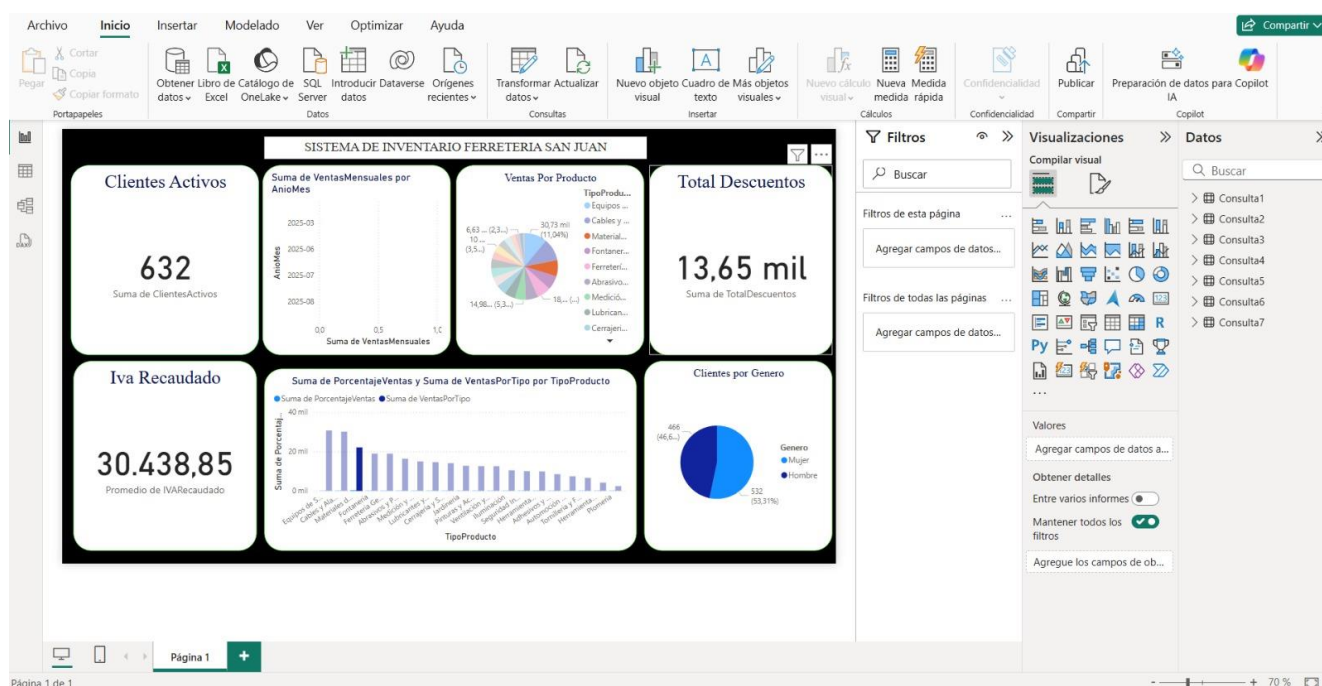
Figura 8*Estructuras empleadas en los sistemas*

3.5 Gráficos obtenidos en los dos sistemas

En los dos sistemas se pueden obtener dashboards interactivos completos relacionando diversos elementos que permiten analizar el giro del negocio desde diferentes perspectivas. Un punto importante que se debe resaltar es que en los dos entornos se pueden obtener los mismos gráficos es decir fueron creados con el mismo fin, la diferencia no resalta en el producto final si no en el proceso de desarrollo, el rendimiento y eficiencia técnica, en este sentido el sistema uno .Net demostró mayor optimización en tiempos de desarrollo e integración con Power BI Pro en contraste con Django que genera tiempos más lentos porque opera bajo un entorno mixto de trabajo.

Figura 9

Gráfico interactivo obtenidos en los dos sistemas

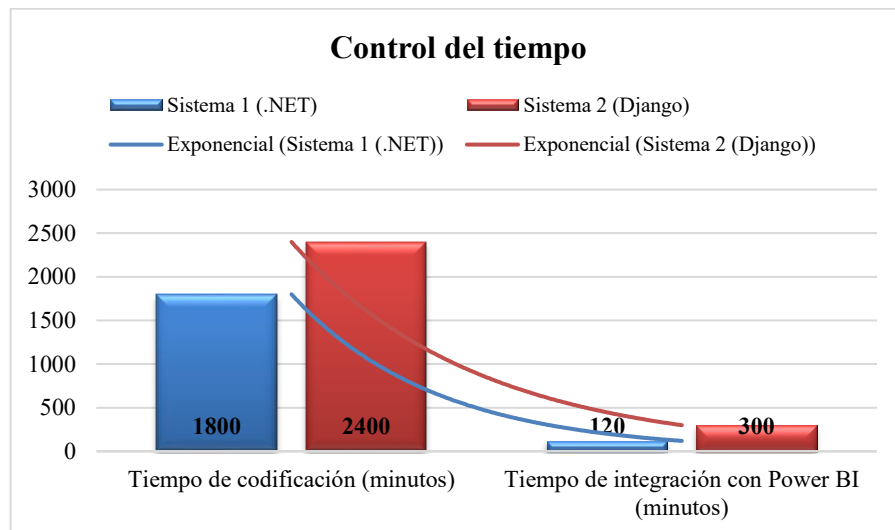


3.6 Comparación de los dos sistemas

Control del tiempo de los sistemas: Se determinó que el sistema 1 Alojado en Windows reflejó menor tiempo en codificación e integración, esto se relaciona con la compatibilidad nativa de Power BI Pro y Windows, en contraste el sistema 2 se toma más tiempo en codificación e integración por la necesidad de generar una estructura que permita la comunicación de Power BI Pro un programa de código cerrado y Linux que maneja código abierto con esto se generó un entorno mixto de trabajo.

Figura 10

Tiempos de conexión de los sistemas



Tipo de integración

Integrar los sistemas con Power BI Pro es el elemento principal de la investigación para determinar la eficiencia de los dos sistemas. En este sentido el tipo de integración influye en el tiempo total de desarrollo y determina el nivel de complejidad de cada proceso, la necesidad de generar una estructura de conexión adicional en el caso del sistema 2 influyó en el tiempo de integración, la escalabilidad y velocidad de despliegue del sistema.

Tabla 3

Tipo de integración de cada sistema

TIPO DE INTEGRACIÓN		
Sistema	Resultado	Observaciones
Sistema 1 (.NET)	Integración directa establecida sin capa intermedia	Compatible nativamente con Power BI Pro
Sistema 2 (Django)	Integración lograda mediante entorno mixto y clúster	Requirió configuración adicional y apertura de puertos

Verificación técnica de los sistemas

Los dos sistemas reflejaron diferencias sustanciales en los diferentes elementos analizados como la productividad, el rendimiento y escalabilidad. Se apreció al sistema 1 alojado en Windows como un sistema más robusto con 5/5 en modularidad, los tiempos de carga fueron más eficientes y con menor latencia del servidor con mayor control y estabilidad del entorno. A diferencia del sistema 2 que presentó menor modularidad y aunque ligeramente el desempeño en tiempo de carga fue mayor, el tiempo de integración incrementó por la configuración de un entorno mixto, los errores detectados fueron 3 a diferencia del otro sistema en el que se detectó solo un error.

Por los resultados se determina que si bien el sistema alojado en Linux es una opción viable que aprovecha el código libre, la integración con otras tecnologías como en el caso Power BI Pro

requiere de mayor tiempo y de elementos técnicos más complejos, esto dificulta el desarrollo y mantenimiento del sistema.

Tabla 4

Verificaciones técnicas de los dos sistemas

VERIFICACIÓN TÉCNICA DE LOS SISTEMAS					
Dimensión	Elemento	Instrumento	Escala de medición	Resultado del Sistema 1	Resultado del Sistema 2
Productividad y escalabilidad	Modularidad del código	Lista de verificación técnica	Escala ordinal	5/5 (estructura completa)	4/5 (estructura parcialmente modular)
Desempeño del sistema	Tiempo de carga del dashboards	Cronómetro / F12 navegador	Escala de razón	3.2 segundos	4.6 segundos
Desempeño del sistema	Tiempo de respuesta del servidor	Postman / Insomnia / consola navegador	Escala de razón	210 ms	320 ms
Desempeño del sistema	Estabilidad del sistema	Registro de incidencias	Escala de razón	1 error	3 errores
Desarrollo	Tiempo de desarrollo	Hoja de control de tiempo	Escala de razón	1800 minutos	2400 minutos
Productividad y escalabilidad	Facilidad de integración con Power BI	Registro de integración	Escala de razón	120 minutos	300 minutos

Detalle de errores detectados

Tabla 5

Detalle de errores

REGISTRO DE ERRORES			
Sistema	Número de errores	Tipo de errores	Resolución aplicada
Sistema 1 (.NET)	1	Error menor en validación	Corrección en validación del formulario
Sistema 2 (Django)	3	Fallos en integración inicial y respuesta lenta	Ajustes en configuración del clúster y optimización de consulta

Al analizar los resultados se precisa que la diferencia entre los dos sistemas no radica en el producto final puesto que visualmente se obtienen los mismos productos, lo interesante es deducir que el tipo de tecnología que se utilice para el desarrollo del sistema impacta en términos de rendimiento y eficiencia del sistema en el que se sustenta. Los dos sistemas lograron integrarse y comunicarse con

Power BI Pro y ofrecer dashboards con experiencia visual semejante donde se reflejan datos claves para el análisis del negocio y la toma de decisiones.

No obstante, el fin del análisis es determinar el sistema que muestra mayor eficiencia en este sentido se encontraron variaciones significativas en cuanto a desarrollo, rendimiento sobre todo al considerar que el sistema 1 .Net se aloja en Windows y se integra de forma directa estableciendo menos tiempo de programación y genera menos errores en las pruebas realizadas situándolo como la opción ideal cuando lo que se requiere es rapidez y facilidad de integración.

El otro sistema Django obtuvo el mismo fin funcional, no obstante, el desarrollo abarcó una mayor complejidad técnica tanto en desarrollo como en integración y arrojó un número mayor de errores, sin embargo, tiene fortalezas como una mayor escalabilidad y es compatible con softwares de código abierto siendo una opción óptima cuando se requiere independencia y personalización tecnológica.

La elección del sistema dependerá de los recursos con los que se cuente, los objetivos y prioridades en que se fundamente el proyecto. En concordancia con (Narváez Vilema et al., 2025) se deben desarrollar sistemas seguros garantizando una integración eficiente con aplicativos como Power BI Pro, es necesario poner atención en la cantidad de información, la naturaleza del negocio a fin de aprovechar las bondades de cada sistema según la disponibilidad y requerimientos con que se emprenda este tipo de proyectos.

Estos resultados se respaldan mediante la aplicación de estadística de t-student de las dos muestras mediante dos variables esenciales para determinar la eficiencia de los dos entornos, el tiempo de desarrollo y la integración con Power BI Pro, donde se puede identificar diferencias relevantes en el criterio de eficiencia. Respecto al tiempo de desarrollo, el Sistema 1 exhibe una media notablemente inferior a la del Sistema 2; esta discrepancia se mostró estadísticamente significativa al aplicar la prueba t de Student ($p < 0,001$) y el intervalo de confianza del 95% se situó completamente por debajo del valor cero, corroborando la ventaja de .NET en esta etapa.

Paralelamente, en el módulo de integración con Power BI, el Sistema 1 presentó un tiempo medio de integración igualmente inferior y estadísticamente significativo ($p < 0,001$). Estos hallazgos sugieren que la adopción del marco .NET no solamente acorta la duración de las fases de desarrollo, sino que también mejora la eficacia de las integraciones, indicando un nivel de productividad que supera al de Django en las circunstancias analizadas.

Tabla 6

T-student tiempo de desarrollo e integración con Power I Pro

Estadísticos descriptivos – Tiempo de desarrollo				
Grupo	N	Media	Desviación típica	Error típico de la media
Sistema 1	5	1800.63	38.07	17.02
Sistema 2	5	2397.74	44.14	19.74

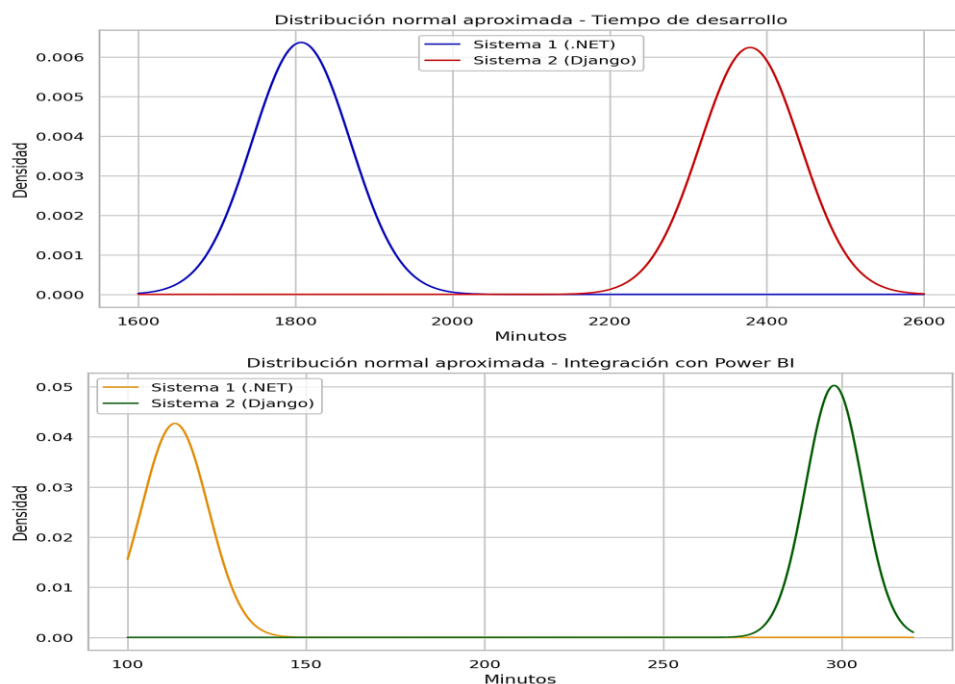
Prueba t para dos muestras – Tiempo de desarrollo						
Variable	T	gl	Sig. (bilateral)	Diferencia de medias	IC 95% Inferior	IC 95% Superior
Tiempo de desarrollo	-14.27	7.99	0	-597.11	-704.2857	-489.9343

Estadísticos descriptivos – Integración con Power BI				
Grupo	N	Media	Desviación típica	Error típico de la media
Sistema 1	5	113.27	9.35	4.18
Sistema 2	5	297.86	7.94	3.55

Prueba t para dos muestras – Integración con Power BI						
Variable	t	gl	Sig. (bilateral)	Diferencia de medias	IC 95% Inferior	IC 95% Superior
Integración con Power BI	-33.66	8	0	-184.5956	-197.3014	-171.8898

Figura 11

Distribuciones normales para el tiempo de desarrollo e integración con Power BI Pro



De acuerdo con los resultados derivados del análisis mediante la prueba t de Student para muestras independientes, se constatan discrepancias estadísticamente significativas entre las plataformas .NET y Django tanto en los intervalos de tiempo de desarrollo como en el proceso de integración con Power BI Pro. En ambos escenarios, los índices de significación estadística se situaron por debajo de 0,001, lo que descarta la hipótesis de que las diferencias observadas puedan explicarse por variación aleatoria. Los menores tiempos registrados por el marco .NET sugieren una mayor eficiencia productiva, circunstancia atribuible a su integración nativa y a las herramientas configuradas para el diseño de interfaces y la conexión con servicios externos como Power BI. En contraste, Django, si bien se presenta como un entorno robusto para el desarrollo web, parece exigir etapas complementarias o bibliotecas de terceros para alcanzar niveles de integración equivalentes, lo que prolonga la duración total del proceso. Tales evidencias permiten postular que, en condiciones controladas, la elección de .NET se erige como la opción más favorable para iniciativas que exigen eficiencia en la creación e integración de dashboards interactivos con Power BI Pro.

Los hallazgos permiten concluir que la hipótesis que postulaba una igualdad en la eficiencia y escalabilidad entre un entorno basado en Django y uno desarrollado en Web Forms para la creación de dashboards interactivos con integración a Power BI Pro no se sostiene. Los análisis estadísticos efectuados evidencian discrepancias significativas en los tiempos de desarrollo y en los plazos de integración; el entorno .NET Web Forms supera al marco de Python en ambas dimensiones. Tal desajuste sugiere que, en el marco de este estudio, Django no logra el mismo rendimiento en

términos de productividad y de celeridad en la elaboración de soluciones interactivas enlazadas a Power BI Pro.

La investigación se armoniza con la investigación desarrollada por (Lombana Pérez, 2024) quien manifiesta que este tipo de sistemas con dashboards interactivos deben ser operables, mostrar información real y útil sobre elementos claves del negocio para mantener un flujo de información en tiempo real, es necesario que en la programación se tenga en cuenta las dificultades técnicas para una configuración eficiente con menos errores y analizar el tipo de tecnología que ofrece mayores ventajas contribuye a futuros proyectos de esta naturaleza con información oportuna que reduce fallas y desperdicio de recursos.

Como punto final cabe recalcar que la investigación se aplicó de forma práctica en el desarrollo y comparación de dos entornos o sistemas con diferente estructura, integrando tecnologías ampliamente utilizadas como Gjango en Linux y .Net en Windows, generando información real sobre la utilidad técnica de cada sistema contribuyendo a que los profesionales cuenten con indicios sobre que tecnología utilizar de acuerdo a las necesidades y enfoques de cada proyecto mediante las especificaciones sobre los aspectos individuales de cada sistema en la aplicación práctica, sin embargo; a pesar de ofrecer datos específicos y reales respecto al desarrollo y rendimiento el estudio presenta limitaciones puesto que el sistema se enfoca en un único negocio y elemento empresarial que en este caso fue el sistema de inventario de una ferretería y los resultados sobre el rendimiento, tiempos de desarrollo, integración con otras tecnologías y errores detectados pueden variar si se emplea mayor cantidad de datos o entornos más complejos, por ello es factible que se en futuras investigaciones se analice los dos sistemas con escenarios de aplicaciones.

3. CONCLUSIÓN

De acuerdo con las pruebas y toma de tiempos que se efectuó se determina que el sistema .Net demostró optimización en tiempos de desarrollo, una integración más rápida y directa con Power BI Pro, fue necesario emplear configuraciones adicionales. Generando una experiencia de programación más fluida situándolo como la opción óptima cuando se requiere rapidez y menor margen de error.

El sistema Django por su parte requirió mayor tiempo de desarrollo e integración por la necesidad de generar una estructura adicional que permitió abrir puerto y comunicar dos tipos de tecnologías una de uso libre como Linux y Power BI Pro que se basa en código cerrado.

Enfatizando en el producto final los dos entornos generan experiencia visual semejante las diferencias sustanciales se generan en la complejidad técnica de programación y conexión con otras

tecnologías. Definitivamente el sistema .Net es ideal para entornos empresariales donde se requiera emplear diversos elementos Microsoft mientras que Django es útil en proyectos libres donde la personalización y escalabilidad es el fin.

Una limitación central de este estudio radica en la dependencia de datos simulados y el tamaño muestral reducido, que puede obstaculizar la extrapolación de los hallazgos a contextos de desarrollo en la práctica. Asimismo, el análisis no integró dimensiones cualitativas, tales como el nivel de experiencia de los desarrolladores ni la complejidad inherente a cada proyecto.

Para investigaciones futuras, se propone la adopción de diseños empíricos que utilicen datos de proyectos en producción, la incorporación de métricas suplementarias —como el consumo de recursos, la mantenibilidad del código y la percepción de los usuarios finales—, y la ampliación de la muestra a una gama más diversa de frameworks y herramientas de visualización empresarial.

CONFLICTO DE INTERESES

No existe conflicto de intereses.

CONTRIBUCIÓN DE AUTORÍA

En concordancia con la taxonomía establecida internacionalmente para la asignación de créditos a autores de artículos científicos (<https://credit.niso.org/>). Los autores declaran sus contribuciones en la siguiente matriz:

	Esquivel Paula Gladys Geoconda	Quisaguano Collaguazo Louis René	Chiguano Lagla Victor Alfonso	Caluña Guzmán Abraham Paul			
Participar activamente en:							
Conceptualización	X	X	X	X			
Análisis formal	X	X	X	X			
Adquisición de fondos	X	X	X	X			
Investigación	X	X	X	X			
Metodología	X	X	X	X			
Administración del proyecto	X	X	X	X			
Recursos	X	X	X	X			
Redacción –borrador original	X	X	X	X			
Redacción –revisión y edición	X	X	X	X			
La discusión de los resultados	X	X	X	X			
Revisión y aprobación de la versión final del trabajo.	X	X	X	X			

REFERENCIAS (APA 7)

Abhishek Parandkar, P. (2024). Enhancing Business Intelligence with Real-Time Sales Dashboards: Integrating SQL Databases and Power BI. *International Journal for Research in Applied Science and Engineering Technology*, 12(5), 1793-1797. <https://doi.org/10.22214/IJRASET.2024.61318>

- Álvarez Medina, D. (2022). *Tratamiento y visualización de datos abiertos sobre la demografía en España* [Tesis de Grado, Escuela Superior de Ingeniería y Tecnología Universidad de la Laguna]. <https://riull.ull.es/xmlui/handle/915/30308>
- Angulo Pinedo, M. (2021). *Diseño y desarrollo de cuadro de mandos para el seguimiento y visualización de datos de asignaturas* [Tesis de Grado, Universidad Politécnica de Madrid]. <https://oa.upm.es/68384/>
- Briones Acuña, L. G., Manríquez Poblete, F. M., & Reyes Roa, M. L. (2024). Caracterización histórica de la concepción de las evaluaciones de ciclo: mecanismos de monitoreo en carreras de pregrado. *Perspectiva Educacional: formación de profesores*, ISSN-e 0718-9729, ISSN 0716-0488, Vol. 63, N°. 2, 2024, págs. 226-250, 63(2), 226-250. <https://doi.org/10.4151/07189729-Vol.63-Iss.2-Art.1562>
- Hidalgo Achig, M., Salguero Núñez, S., Sandoval Cárdenas, M., & Iza Garnica, J. (2021). Teleeducación: Brecha digital una realidad palpable: una mirada desde la comunidad educativa. *Horizontes Revista de Investigación en Ciencias de la Educación*, 5(21), 216-232. <https://doi.org/10.33996/REVISTAHORIZONTES.V5I21.298>
- IT Madrid. (2021, mayo 19). *Qué es Inteligencia de Negocios (Business Intelligence) | ITMadrid Digital School*. <https://www.itmadrid.com/que-es-inteligencia-de-negocios-business-intelligence/>
- Lombana Perez, W. (2021). *Herramienta de Análisis de Desarrollo de Proyectos: Visualización de Tiempos y Eficiencia a través de Power BI*. [Tesis de Grado, UNAD]. <https://repository.unad.edu.co/handle/10596/63754>
- Lombana Pérez, W. (2024). *Proyecto de grado aplicado Herramienta de Análisis de Desarrollo de Proyectos: Visualización de Tiempos y Eficiencia a través de Power BI* [Universidad Nacional Abierta y a Distancia UNAD]. <https://repository.unad.edu.co/bitstream/handle/10596/63754/wlombanap.pdf?sequence=3&isAllowed=y>
- Morales Sevilla, C. N. (2021). *Modelo de seguimiento y monitoreo de proyectos en la etapa de ejecución a través de una aplicación web utilizando criterios de usabilidad* [Tesis de Posgrado, Universidad Técnica del Norte]. <https://repositorio.utn.edu.ec/handle/123456789/7609>
- Narváez Vilema, M., Calderón Orozco, J., & Peña Guevara, B. (2025). *Aplicativo web para el monitoreo del proceso de vinculación de la Universidad Nacional de Chimborazo utilizando la tecnología .Net*. [Tesis de Grado, Universidad Nacional de Chimborazo]. <http://dspace.unach.edu.ec/handle/51000/15279>
- Pacheco Huanca, R., & Dionicio Asto, M. (2024). *Sistema Integrado de Toma de Decisiones para la Gestión Comercial en el Mercado Peruano utilizando Inteligencia de Negocios y Power Bi Pro* [Tesis de Grado, Universidad Peruana de Ciencias Aplicadas (UPC)].

https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/683868/Dionicio_AM.pdf?sequence=1&isAllowed=y

Quishpe Zapata, A. (2024). *Desarrollo de Sistema de Inteligencia de Negocios par el manejo de Indicadores de Tickets de Soporte*. [Tesis de Grado, Escuela Politécnica Nacional].
<http://bibdigital.epn.edu.ec/handle/15000/25519>