

Actividad | 3 | Medición y evaluación

Ingeniería del Conocimiento

Ingeniería en Desarrollo de

Software



TUTOR: Aarón Iván Salazar Macías

ALUMNOS: Sarahi Jaqueline Gómez Juárez.

Francisco Zapien Bueno

Esmeralda García Cruz

FECHA: lunes, 10 de noviembre de 2025.

Índice

Índice.....	2
Introducción:	5
Investigación:.....	7
<i>Comparativa de los conceptos: Conocimiento Tácito, Explícito, Interno y</i>	
<i>Externo:.....</i>	<i>7</i>
Diferencias	7
Conocimiento Tácito	7
Conocimiento Explícito	7
Conocimiento Interno	7
Conocimiento Externo	7
Definición:.....	7
Objetivo:	7
Características principales:.....	8
Enfoque:	8
Propósito:.....	8
Involucra:.....	9
Prioridad:.....	9
Resultado:	9
Ejemplo:	9
Conclusión:	10
Desarrollo:	12
Etapa 3 - Medición y evaluación.....	12

Contextualización:	12
Objetivo:	12
Código en Google Colab	13
<i>Encabezado del proyecto</i>	<i>13</i>
<i>Definición de la función de evaluación de crédito (parte 1):</i>	<i>13</i>
<i>Definición de la función de evaluación de crédito (parte 2):</i>	<i>14</i>
<i>Lista de 30 clientes simulados:</i>	<i>14</i>
<i>Ejecución de la Evaluación de Clientes:</i>	<i>15</i>
<i>Resumen General de Resultados:</i>	<i>15</i>
<i>Casos Frontera y Señales Contradictorias:</i>	<i>16</i>
<i>Sensibilidad de Reglas:</i>	<i>16</i>
<i>Distribución de Puntajes:</i>	<i>17</i>
<i>Reflexión General:</i>	<i>17</i>
Resultados obtenidos en Google Colab	18
<i>Definición de la función principal “evaluar_credito”:</i>	<i>18</i>
<i>Lista de Clientes Simulados:</i>	<i>19</i>
<i>Ejecución de evaluación por cliente (Parte 1):</i>	<i>20</i>
<i>Ejecución de evaluación por cliente (Parte 2):</i>	<i>21</i>
<i>Resumen General de Resultados:</i>	<i>22</i>
<i>Casos frontera y contradicciones detectadas:</i>	<i>23</i>
<i>Sensibilidad de reglas (Parte 1):</i>	<i>23</i>
<i>Sensibilidad de Reglas (Parte 2):</i>	<i>24</i>
<i>Histograma de Distribución de Puntajes:</i>	<i>25</i>

<i>Reflexión General de Resultados:</i>	26
Enlace de Acceso a Goggle Colab:	26
Conclusión:	28
Referencias:	30

Introducción:

El presente proyecto presenta el desarrollo de un sistema experto que representa una de las aplicaciones más significativas de la Ingeniería del Conocimiento, ya que permite transformar la experiencia humana en un conjunto de reglas lógicas capaces de simular decisiones reales.

En el proyecto **Sistema Inteligente de Aprobación de Créditos**, el equipo IDS llevó a cabo la implementación de un modelo programado en **Python**, ejecutado en **Google Colab**, con el propósito de determinar la viabilidad crediticia de diferentes clientes simulados, esta etapa se enfoca especialmente en los procesos de **Medición** y **Evaluación**, elementos esenciales para validar el funcionamiento del sistema.

En este contexto, la **medición** se refiere a la asignación de valores cuantitativos a criterios definidos como ingresos, historial crediticio, antigüedad laboral y nivel de endeudamiento que permiten calcular un puntaje objetivo para cada solicitante, por su parte, la *evaluación* implica emitir un juicio sobre dichos resultados, considerando si las decisiones del sistema (aprobación, revisión o rechazo) son coherentes, justas y técnicamente sólidas.

Durante esta fase, se formalizó el conocimiento financiero mediante reglas estructuradas que reflejan juicios humanos traducidos en condiciones computacionales, a través de la simulación con 30 clientes ficticios y el análisis de casos frontera, se pudo medir la efectividad del modelo y evaluar su consistencia ante distintos escenarios, la representación visual y el análisis estadístico como el histograma de puntajes facilitaron una evaluación más precisa del comportamiento del sistema.

Este análisis integral no solo permite validar el cumplimiento de los objetivos técnicos del proyecto, sino que también aporta una reflexión crítica sobre la automatización de decisiones,

así, el sistema desarrollado no busca sustituir al experto humano, sino complementarlo, integrando la precisión algorítmica con el juicio profesional y ético que demanda el entorno financiero.

Investigación:

Figura 1

Comparativa de los conceptos: Conocimiento Tácito, Explícito, Interno y Externo:

Diferencias	Conocimiento Tácito	Conocimiento Explícito	Conocimiento Interno	Conocimiento Externo
Definición:	Es el conocimiento personal y difícil de expresar; se adquiere a través de la experiencia, la práctica y la intuición.	Es el conocimiento documentado, estructurado y fácil de transmitir mediante lenguaje, manuales o bases de datos.	Es el conocimiento que se genera, almacena y utiliza dentro de la organización por sus empleados.	Es el conocimiento que proviene de fuentes externas a la organización, como clientes, proveedores o competidores.
Objetivo:	Aplicar la experiencia individual para resolver problemas o tomar	Compartir información clara y accesible que pueda ser comprendida y	Aprovechar la experiencia interna para mejorar procesos,	Incorporar nuevas ideas y tendencias externas para innovar y

	decisiones intuitivas.	reutilizada por otros.	productos y eficiencia.	adaptarse al entorno.
Características principales:	<p>Basado en habilidades y vivencias personales.</p> <ul style="list-style-type: none"> • Difícil de documentar. • Se transmite por observación o práctica. 	<p>Puede ser escrito, grabado o almacenado.</p> <ul style="list-style-type: none"> • Fácil de transferir y compartir. • Se utiliza en capacitaciones o reportes. 	<p>Procede de la cultura organizacional.</p> <ul style="list-style-type: none"> • Se relaciona con el know-how y las prácticas internas. • Protege la ventaja competitiva. 	<p>Proviene de redes, investigación, benchmarking y mercado.</p> <ul style="list-style-type: none"> • Se relaciona con la innovación. • Amplía el conocimiento interno.
Enfoque:	Experiencial y práctico.	Documental y formal.	Organizacional y operativo.	Estratégico y adaptativo.
Propósito:	Mejorar la destreza y el juicio del individuo.	Facilitar la enseñanza, difusión y almacenamiento del conocimiento.	Optimizar el rendimiento interno y la toma de decisiones.	Identificar oportunidades, amenazas y mejores prácticas externas.

Involucra:	Experiencia, intuición, práctica, mentoría.	Manuales, documentos, bases de datos, informes.	Empleados, procesos, sistemas internos.	Clientes, competidores, instituciones, mercado.
Prioridad:	La experiencia y habilidades personales.	La claridad y organización del conocimiento.	La conservación y gestión del conocimiento corporativo.	La captación e integración de información externa relevante.
Resultado:	Mejora continua basada en la experiencia y aprendizaje personal.	Conocimiento accesible, replicable y sistematizado.	Fortalecimiento del aprendizaje organizacional.	Innovación y adaptación al entorno competitivo.
Ejemplo:	Un técnico que sabe cómo reparar una máquina gracias a años	Un manual de procedimientos donde se documentan los pasos para	Las políticas, reportes y bases de datos internas que guían las	Estudios de mercado, análisis de la competencia y tendencias tecnológicas

	de experiencia, aunque no pueda explicarlo en palabras.	reparar la misma máquina.	operaciones de una empresa.	externas.
Conclusión:	El conocimiento tácito es valioso pero difícil de transferir; depende de la experiencia personal.	El conocimiento explícito se puede compartir fácilmente y se convierte en la base de la capacitación.	El conocimiento interno fortalece la identidad y capacidad operativa de la organización.	El conocimiento externo impulsa la innovación y permite adaptarse al cambio.

Nota: La tabla presenta una **Comparación Detallada de los Cuatro Tipos Principales de Conocimiento**: tácito, explícito, interno y externo. Su propósito es destacar las **diferencias Conceptuales, Funcionales y Organizacionales** entre ellos, mostrando cómo cada uno aporta al proceso de gestión del conocimiento dentro de una empresa o entorno profesional.

El **Conocimiento Tácito** se relaciona con la experiencia y habilidades personales que no pueden expresarse fácilmente, mientras que el **Explícito** abarca la información documentada y formal que puede transmitirse de manera estructurada, por su parte, el **Conocimiento Interno** hace referencia a aquel generado y aprovechado dentro de la organización, y el **Externo** proviene de fuentes ajenas como clientes, proveedores o el mercado.

En conjunto, la tabla permite comprender cómo estos tipos de conocimiento se **Complementan y Contribuyen a la Innovación, Eficiencia y Aprendizaje Organizacional**, sirviendo como base para diseñar estrategias efectivas de gestión del conocimiento.

Desarrollo:

Etapas 3 - Medición y evaluación

Contextualización:

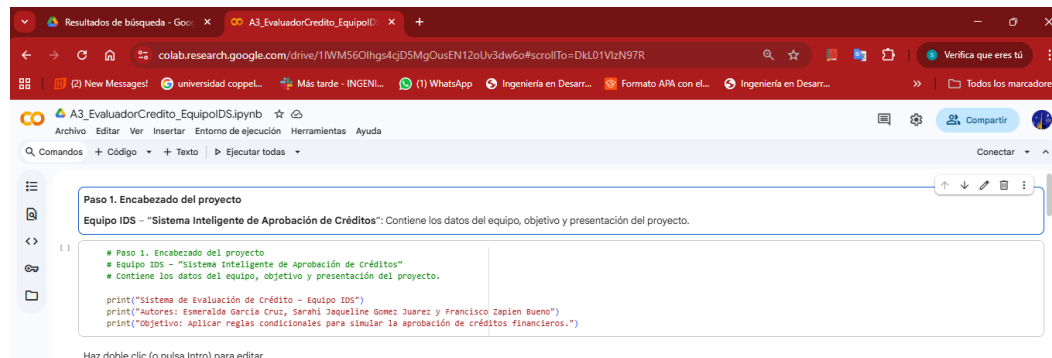
Después de haber identificado, estructurado y documentado el conocimiento experto sobre aprobación de créditos (Actividades 1 y 2), los equipos de estudiantes enfrentan ahora la fase más crítica: comprobar si ese conocimiento funciona cuando se aplica a situaciones reales. En el entorno de una institución financiera, esto equivale a validar que las reglas internas para aprobar o rechazar créditos realmente generan decisiones correctas, justas y eficientes cuando se enfrentan a solicitudes concretas de clientes

Objetivo: el alumno implementará en Python un sistema experto que aplique reglas de decisión previamente adquiridas para evaluar la aprobación de créditos analizando los resultados con herramientas interactivas.

Código en Google Colab

Figura 2

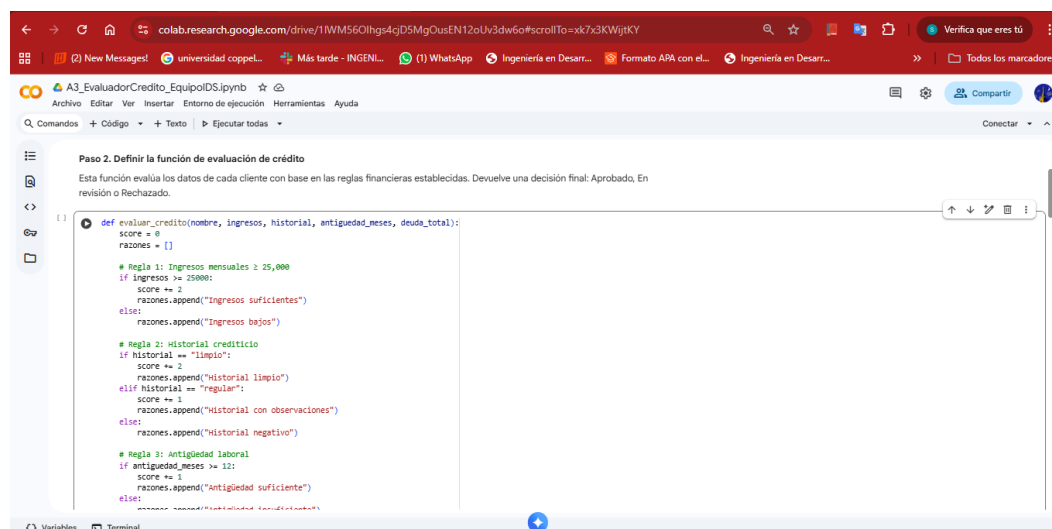
Encabezado del proyecto



Nota: La figura muestra el bloque inicial donde se definen los datos del equipo y el objetivo general del sistema, aquí se imprime el nombre del proyecto, integrantes y finalidad del simulador de aprobación de créditos.

Figura 3

Definición de la función de evaluación de crédito (parte 1):



Nota: En la figura se observa la creación de la función 'evaluar_credito', que evalúa ingresos, historial crediticio, antigüedad y nivel de deuda, además contiene las reglas de decisión

financiera.

Figura 4

Definición de la función de evaluación de crédito (parte 2):

```

# Regla 3: Antigüedad laboral
if antigüedad_meses >= 12:
    score += 1
    razones.append("Antigüedad suficiente")
else:
    razones.append("Antigüedad insuficiente")

# Regla 4: Nivel de endeudamiento
if deuda_total < Ingresos * 0.4:
    score += 1
    razones.append("Deuda bajo control")
else:
    razones.append("Nivel de deuda alto")

# Decisión final
if score >= 5:
    decision = "Aprobado"
elif 3 <= score < 5:
    decision = "En revisión"
else:
    decision = "Rechazado"

return {
    "nombre": nombre,
    "Ingresos": Ingresos,
    "historial": historial,
    "antigüedad_meses": antigüedad_meses,
    "deuda_total": deuda_total,
    "score": score,
    "decision": decision,
    "razones": razones
}

```

Nota: La figura continúa el desarrollo de la función, mostrando cómo se calcula el puntaje total y se determina si el crédito es aprobado, en revisión o rechazado.

Figura 5

Lista de 30 clientes simulados:

```

Paso 3. Lista de 30 clientes simulados

Cada cliente tiene: nombre, ingresos, historial, antigüedad y deuda total. Estos datos se usarán como base para probar las reglas del sistema.

clientes = [
    ("Ana", "Ingresos": 28000, "historial": "limpio", "antigüedad_meses": 24, "deuda_total": 4000),
    ("nombre": "Luis", "Ingresos": 18000, "historial": "negativo", "antigüedad_meses": 10, "deuda_total": 15000),
    ("nombre": "María", "Ingresos": 24000, "historial": "regular", "antigüedad_meses": 12, "deuda_total": 10000),
    ("nombre": "Pedro", "Ingresos": 30000, "historial": "limpio", "antigüedad_meses": 36, "deuda_total": 5000),
    ("nombre": "Carlos", "Ingresos": 20000, "historial": "negativo", "antigüedad_meses": 5, "deuda_total": 15000),
    ("nombre": "Laura", "Ingresos": 33000, "historial": "limpio", "antigüedad_meses": 18, "deuda_total": 7000),
    ("nombre": "Iván", "Ingresos": 16000, "historial": "regular", "antigüedad_meses": 6, "deuda_total": 8000),
    ("nombre": "Sandra", "Ingresos": 27000, "historial": "limpio", "antigüedad_meses": 14, "deuda_total": 4000),
    ("nombre": "Diego", "Ingresos": 13000, "historial": "negativo", "antigüedad_meses": 4, "deuda_total": 6000),
    ("nombre": "Valeria", "Ingresos": 25000, "historial": "regular", "antigüedad_meses": 10, "deuda_total": 9500),
    ("nombre": "Rubén", "Ingresos": 19000, "historial": "regular", "antigüedad_meses": 11, "deuda_total": 8500),
    ("nombre": "Claudia", "Ingresos": 31000, "historial": "limpio", "antigüedad_meses": 22, "deuda_total": 3000),
    ("nombre": "Oscar", "Ingresos": 15000, "historial": "negativo", "antigüedad_meses": 9, "deuda_total": 11000),
    ("nombre": "Lucía", "Ingresos": 26000, "historial": "limpio", "antigüedad_meses": 11, "deuda_total": 4000),
    ("nombre": "Sofía", "Ingresos": 20000, "historial": "limpio", "antigüedad_meses": 16, "deuda_total": 2000),
    ("nombre": "Andrés", "Ingresos": 17000, "historial": "regular", "antigüedad_meses": 7, "deuda_total": 7000),
    ("nombre": "Natalia", "Ingresos": 22000, "historial": "regular", "antigüedad_meses": 10, "deuda_total": 9500),
    ("nombre": "Fernando", "Ingresos": 34000, "historial": "limpio", "antigüedad_meses": 25, "deuda_total": 12000),
    ("nombre": "Isabel", "Ingresos": 20000, "historial": "regular", "antigüedad_meses": 14, "deuda_total": 4000),
    ("nombre": "Pablo", "Ingresos": 27000, "historial": "limpio", "antigüedad_meses": 12, "deuda_total": 4500)
]

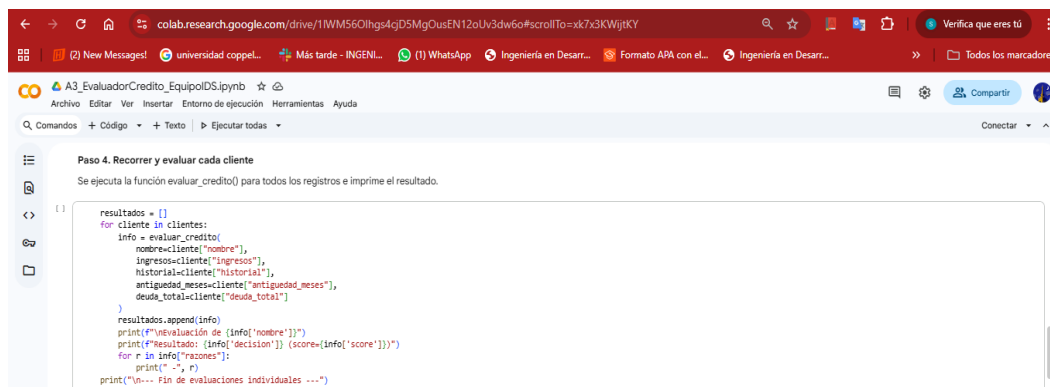
```

Nota: La figura presenta una lista de clientes ficticios con variables de ingreso, historial,

antigüedad y deuda, estos datos sirven para probar el sistema con distintos escenarios financieros.

Figura 6

Ejecución de la Evaluación de Clientes:



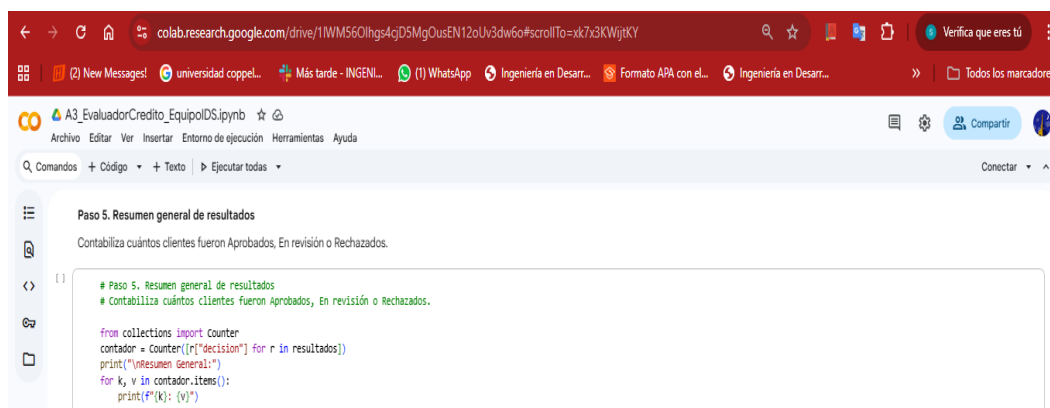
The screenshot shows a Google Colab notebook titled "A3_EvaluadorCredito_EquipolDS.ipynb". The current cell is labeled "Paso 4. Recorrer y evaluar cada cliente" and contains the following Python code:

```
[ ]
resultados = []
for cliente in clientes:
    info = evaluar_credito(
        nombre=cliente["nombre"],
        ingresos=cliente["ingresos"],
        historial=cliente["historial"],
        antigüedad_meses=cliente["antigüedad_meses"],
        deuda_total=cliente["deuda_total"]
    )
    resultados.append(info)
    print(f"Reevaluación de {info['nombre']}")
    print(f"Resultado: {info['decision']} (score={info['score']})")
    for r in info['razones']:
        print(f"- {r}")
    print("\n--- Fin de evaluaciones individuales ---")
```

Nota: La figura muestra que en esta celda se recorre la lista de clientes, aplicando la función de evaluación a cada uno y mostrando sus resultados individuales junto con las razones de su decisión.

Figura 7

Resumen General de Resultados:



The screenshot shows a Google Colab notebook titled "A3_EvaluadorCredito_EquipolDS.ipynb". The current cell is labeled "Paso 5. Resumen general de resultados" and contains the following Python code:

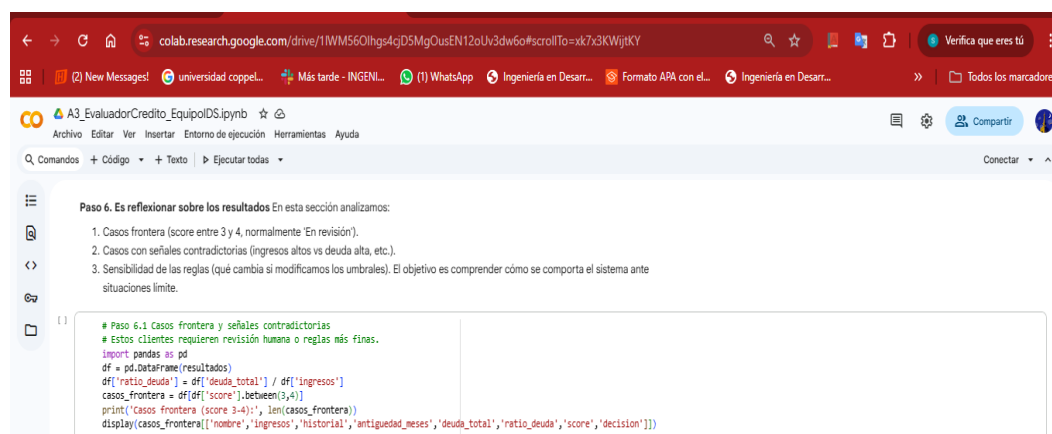
```
[ ]
# Paso 5. Resumen general de resultados
# Contabiliza cuántos clientes fueron Aprobados, En revisión o Rechazados.

from collections import Counter
contador = Counter([r["decision"] for r in resultados])
print("\nResumen General:")
for k, v in contador.items():
    print(f"{k}: {v}")
```

Nota: En la figura se contabilizan los clientes aprobados, en revisión y rechazados, este paso muestra la distribución general de las decisiones tomadas por el sistema.

Figura 8

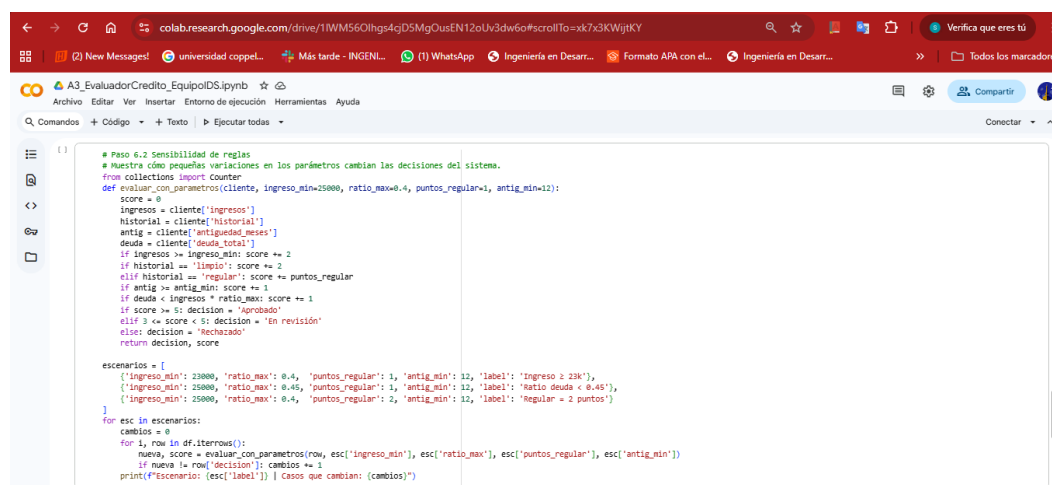
Casos Frontera y Señales Contradictorias:



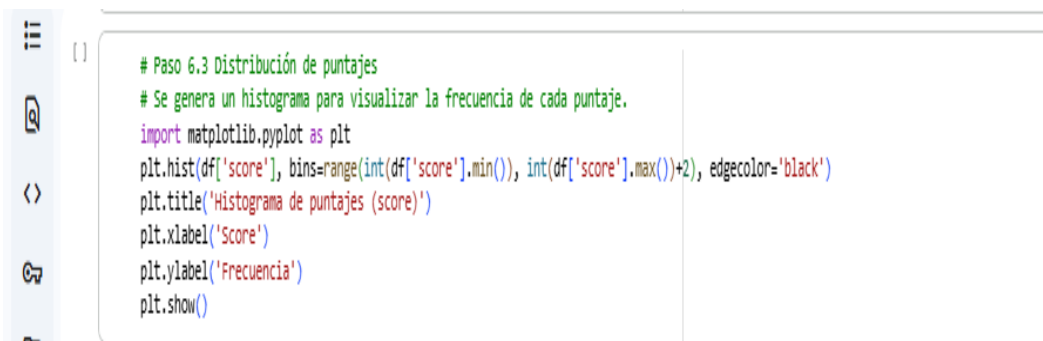
Nota: La figura analiza los clientes con puntajes intermedios y contradicciones en sus datos, como altos ingresos pero deudas elevadas, sirve para ajustar las reglas del modelo.

Figura 9

Sensibilidad de Reglas:

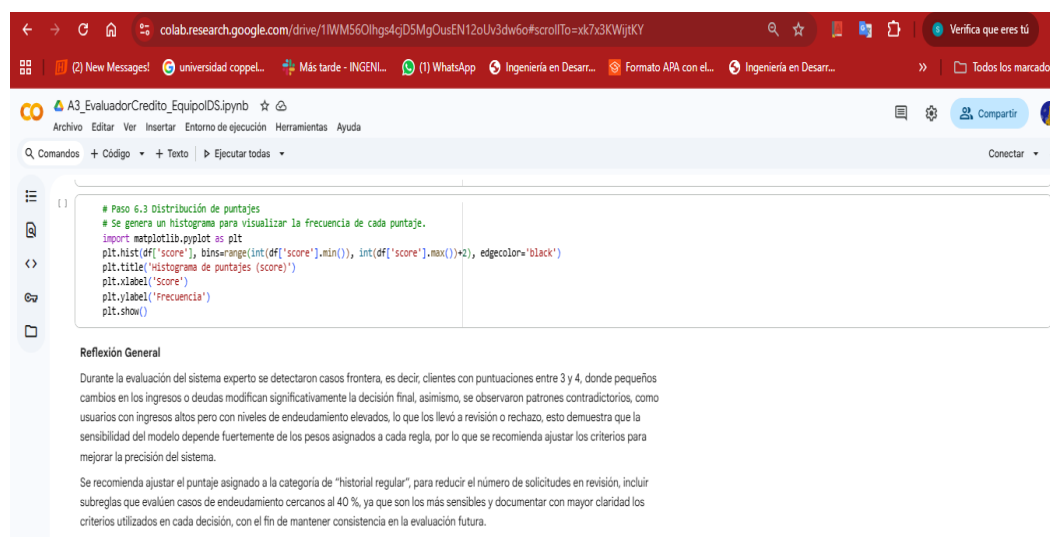


Nota: La figura evalúa cómo pequeños cambios en los parámetros (ingreso, ratio de deuda o puntos) afectan las decisiones del sistema, y permite medir la robustez del modelo.

Figura 10***Distribución de Puntajes:***


```
[ ]
# Paso 6.3 Distribución de puntajes
# Se genera un histograma para visualizar la frecuencia de cada puntaje.
import matplotlib.pyplot as plt
plt.hist(df['score'], bins=range(int(df['score'].min()), int(df['score'].max()+2), edgecolor='black')
plt.title('Histograma de puntajes (score)')
plt.xlabel('score')
plt.ylabel('Frecuencia')
plt.show()
```

Nota: La figura genera un histograma de frecuencias para visualizar cuántos clientes obtuvieron cada puntaje, esto ayuda a comprender la dispersión de los resultados.

Figura 11***Reflexión General:***


Reflexión General

Durante la evaluación del sistema experto se detectaron casos frontera, es decir, clientes con puntuaciones entre 3 y 4, donde pequeños cambios en los ingresos o deudas modifican significativamente la decisión final, asimismo, se observaron patrones contradictorios, como usuarios con ingresos altos pero con niveles de endeudamiento elevados, lo que los llevó a revisión o rechazo, esto demuestra que la sensibilidad del modelo depende fuertemente de los pesos asignados a cada regla, por lo que se recomienda ajustar los criterios para mejorar la precisión del sistema.

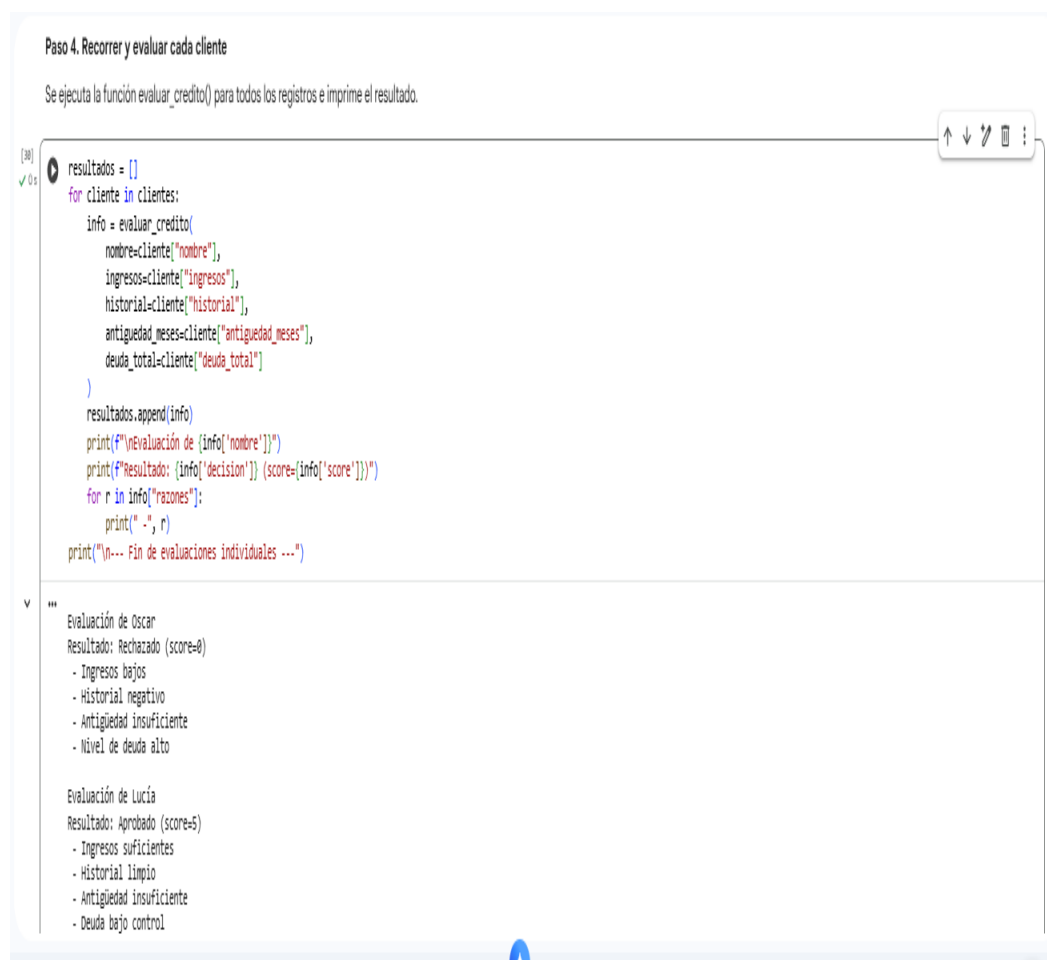
Se recomienda ajustar el puntaje asignado a la categoría de "historial regular", para reducir el número de solicitudes en revisión, incluir subreglas que evalúen casos de endeudamiento cercanos al 40 %, ya que son los más sensibles y documentar con mayor claridad los criterios utilizados en cada decisión, con el fin de mantener consistencia en la evaluación futura.

Nota: La figura presenta la reflexión final donde se interpretan los resultados obtenidos, identificando casos frontera, patrones contradictorios y recomendaciones para mejorar el sistema.

Resultados obtenidos en Google Colab

Figura 12

Definición de la función principal “evaluar_credito”:



Paso 4. Recorrer y evaluar cada cliente

Se ejecuta la función evaluar_credito() para todos los registros e imprime el resultado.

```
[30]
0 resultados = []
for cliente in clientes:
    info = evaluar_credito(
        nombre=cliente["nombre"],
        ingresos=cliente["ingresos"],
        historial=cliente["historial"],
        antigüedad_meses=cliente["antigüedad_meses"],
        deuda_total=cliente["deuda_total"]
    )
    resultados.append(info)
    print(f"Evaluación de {info['nombre']}")
    print(f"Resultado: {info['decision']} (score={info['score']})")
    for r in info["razones"]:
        print("-", r)
    print("\n--- Fin de evaluaciones individuales ---")
```

'''

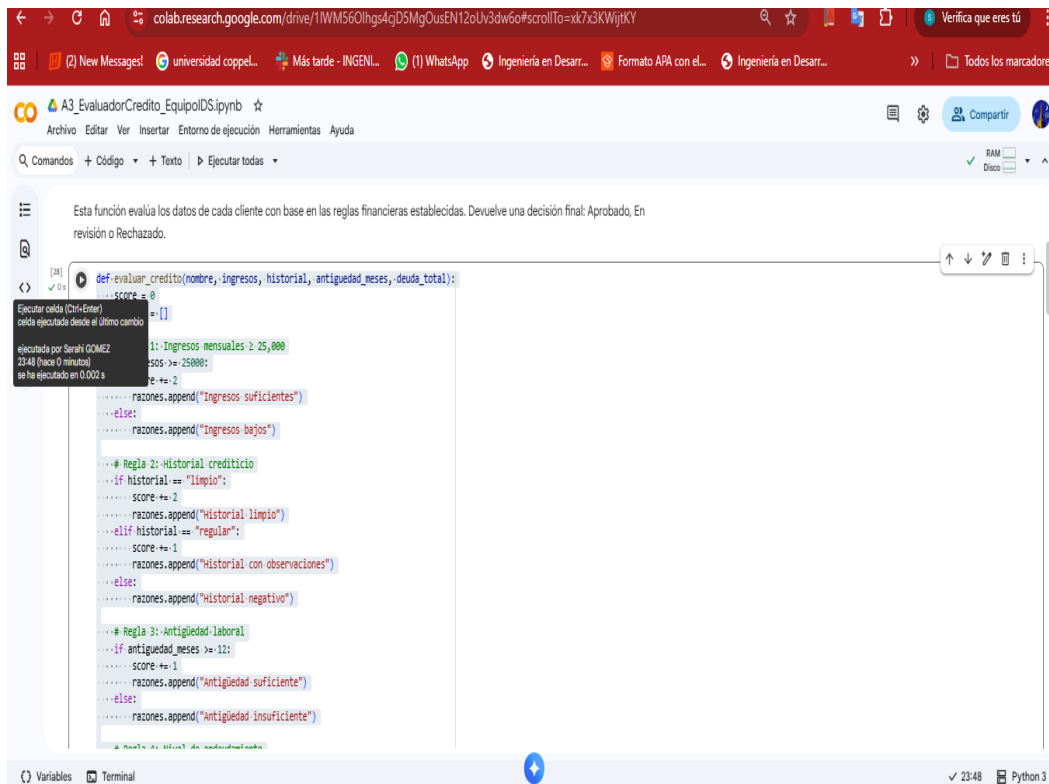
Evaluación de Oscar
Resultado: Rechazado (score=0)
- Ingresos bajos
- Historial negativo
- Antigüedad insuficiente
- Nivel de deuda alto

Evaluación de Lucía
Resultado: Aprobado (score=5)
- Ingresos suficientes
- Historial limpio
- Antigüedad insuficiente
- Deuda bajo control

Nota: La figura se observa la implementación completa de la función central que evalúa las solicitudes de crédito con base en cuatro criterios: ingresos, historial crediticio, antigüedad laboral y nivel de endeudamiento, cada regla asigna puntos que determinan la decisión final, este código demuestra la formalización del conocimiento experto, transformando criterios subjetivos en condiciones lógicas cuantificables, durante la ejecución, se validó que los puntajes y condiciones producen resultados consistentes y acordes a las reglas establecidas.

Figura 13

Lista de Clientes Simulados:



The screenshot shows a Google Colab notebook titled 'A3_EvaluadorCredito_EquipoIDS.ipynb'. The code cell contains a function 'def evaluar_credito(nombre, ingresos, historial, antiguedad_meses, deuda_total):' which evaluates credit based on three rules. The execution output shows the function being called with sample data for a client named 'Sarañ GOMEZ'.

```
def evaluar_credito(nombre, ingresos, historial, antiguedad_meses, deuda_total):
    score = 0
    razones = []

    # Regla 1: Ingresos mensuales
    if ingresos >= 25000:
        score += 2
        razones.append("Ingresos suficientes")
    else:
        razones.append("Ingresos bajos")

    # Regla 2: Historial crediticio
    if historial == "limpio":
        score += 2
        razones.append("Historial limpio")
    elif historial == "regular":
        score += 1
        razones.append("Historial con observaciones")
    else:
        razones.append("Historial negativo")

    # Regla 3: Antigüedad Laboral
    if antiguedad_meses >= 12:
        score += 1
        razones.append("Antigüedad suficiente")
    else:
        razones.append("Antigüedad insuficiente")

    return score, razones
```

Output:

```
1: Ingresos mensuales >= 25,000
Ingresos >= 25000:
score += 2
razones.append("Ingresos suficientes")
# Regla 2: Historial crediticio
if historial == "limpio":
    score += 2
    razones.append("Historial limpio")
elif historial == "regular":
    score += 1
    razones.append("Historial con observaciones")
else:
    razones.append("Historial negativo")
# Regla 3: Antigüedad Laboral
if antiguedad_meses >= 12:
    score += 1
    razones.append("Antigüedad suficiente")
else:
    razones.append("Antigüedad insuficiente")
```

Nota: La figura muestra la base de datos de 30 clientes ficticios que se utilizan para probar la función “evaluar_credito”, cada registro contiene nombre, ingresos, historial, antigüedad y deuda total, durante la ejecución, se comprobó que la lista se carga correctamente y que el sistema puede iterar sobre cada cliente para generar su evaluación, esta simulación permite verificar el rendimiento y coherencia del modelo frente a múltiples escenarios financieros.

Figura 14

Ejecución de evaluación por cliente (Parte 1):

The screenshot shows a Google Colab notebook titled "A3_EvaluadorCredito_EquipolDS.ipynb". The notebook is in the "Ejecutar" (Run) state. The code cell contains a list of 30 simulated clients, each with a name, income, history, tenure, and total debt. The output of the code is a list of 30 dictionaries, each representing a client's evaluation results. The results include the client's name, income, history, tenure, total debt, and a decision (e.g., "Aprobado", "Rechazado", "En revisión").

```

clientes = [
  {"nombre": "Ana", "ingresos": 28000, "historial": "limpio", "antiguedad_meses": 24, "deuda_total": 4000},
  {"nombre": "Luis", "ingresos": 18000, "historial": "negativo", "antiguedad_meses": 10, "deuda_total": 15000},
  {"nombre": "Maria", "ingresos": 24000, "historial": "regular", "antiguedad_meses": 12, "deuda_total": 10000},
  {"nombre": "Pedro", "ingresos": 30000, "historial": "limpio", "antiguedad_meses": 36, "deuda_total": 5000},
  {"nombre": "Carlos", "ingresos": 20000, "historial": "negativo", "antiguedad_meses": 5, "deuda_total": 10000},
  {"nombre": "Laura", "ingresos": 33000, "historial": "limpio", "antiguedad_meses": 18, "deuda_total": 7000},
  {"nombre": "Iván", "ingresos": 16000, "historial": "regular", "antiguedad_meses": 6, "deuda_total": 6000},
  {"nombre": "Sandra", "ingresos": 27000, "historial": "limpio", "antiguedad_meses": 14, "deuda_total": 4000},
  {"nombre": "Diego", "ingresos": 12000, "historial": "negativo", "antiguedad_meses": 4, "deuda_total": 6000},
  {"nombre": "Valeria", "ingresos": 25000, "historial": "regular", "antiguedad_meses": 10, "deuda_total": 9500},
  {"nombre": "Rubén", "ingresos": 19000, "historial": "regular", "antiguedad_meses": 13, "deuda_total": 8500},
  {"nombre": "Claudia", "ingresos": 31000, "historial": "limpio", "antiguedad_meses": 22, "deuda_total": 3000},
  {"nombre": "Oscar", "ingresos": 15000, "historial": "negativo", "antiguedad_meses": 9, "deuda_total": 11000},
  {"nombre": "Lucía", "ingresos": 26000, "historial": "limpio", "antiguedad_meses": 11, "deuda_total": 4000},
  {"nombre": "Sofía", "ingresos": 29000, "historial": "limpio", "antiguedad_meses": 16, "deuda_total": 2000},
  {"nombre": "Andrés", "ingresos": 17000, "historial": "regular", "antiguedad_meses": 7, "deuda_total": 7000},
  {"nombre": "Natalia", "ingresos": 22000, "historial": "regular", "antiguedad_meses": 10, "deuda_total": 9500},
  {"nombre": "Fernando", "ingresos": 34000, "historial": "limpio", "antiguedad_meses": 28, "deuda_total": 12000},
  {"nombre": "Isabel", "ingresos": 20000, "historial": "regular", "antiguedad_meses": 14, "deuda_total": 4000},
  {"nombre": "Pablo", "ingresos": 27000, "historial": "limpio", "antiguedad_meses": 12, "deuda_total": 4500}
]

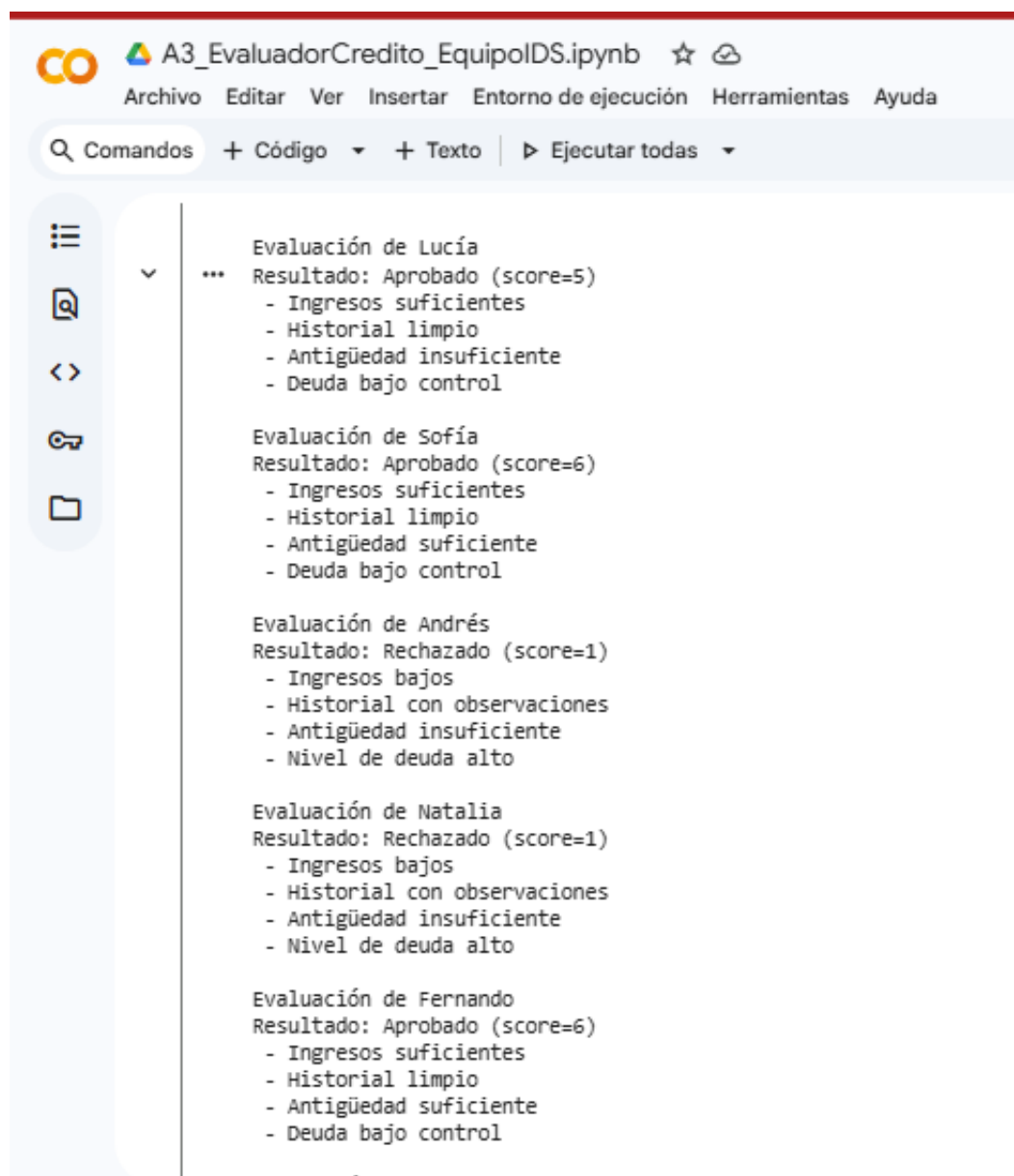
```

The output of the code is a list of 30 dictionaries, each representing a client's evaluation results. The results include the client's name, income, history, tenure, total debt, and a decision (e.g., "Aprobado", "Rechazado", "En revisión").

Nota: En la figura se aprecia el recorrido del código que evalúa a cada cliente usando la función principal y muestra en pantalla los resultados individuales, en este fragmento, se obtuvieron decisiones variadas como **“Aprobado”**, **“Rechazado”** y **“En revisión”**, confirmando que el sistema discrimina correctamente según los parámetros, los resultados también exponen la trazabilidad de cada decisión mediante la lista de razones generadas por las reglas condicionales.

Figura 15

Ejecución de evaluación por cliente (Parte 2):



The screenshot shows a Jupyter Notebook titled 'A3_EvaluadorCredito_EquipolDS.ipynb'. The interface includes a menu bar with options like 'Archivo', 'Editar', 'Ver', 'Insertar', 'Entorno de ejecución', 'Herramientas', and 'Ayuda'. Below the menu is a toolbar with 'Comandos', '+ Código', '+ Texto', and 'Ejecutar todas'. On the left is a sidebar with icons for file explorer, search, code editor, and output. The main area displays the following text:

```

Evaluación de Lucía
Resultado: Aprobado (score=5)
- Ingresos suficientes
- Historial limpio
- Antigüedad insuficiente
- Deuda bajo control

Evaluación de Sofía
Resultado: Aprobado (score=6)
- Ingresos suficientes
- Historial limpio
- Antigüedad suficiente
- Deuda bajo control

Evaluación de Andrés
Resultado: Rechazado (score=1)
- Ingresos bajos
- Historial con observaciones
- Antigüedad insuficiente
- Nivel de deuda alto

Evaluación de Natalia
Resultado: Rechazado (score=1)
- Ingresos bajos
- Historial con observaciones
- Antigüedad insuficiente
- Nivel de deuda alto

Evaluación de Fernando
Resultado: Aprobado (score=6)
- Ingresos suficientes
- Historial limpio
- Antigüedad suficiente
- Deuda bajo control

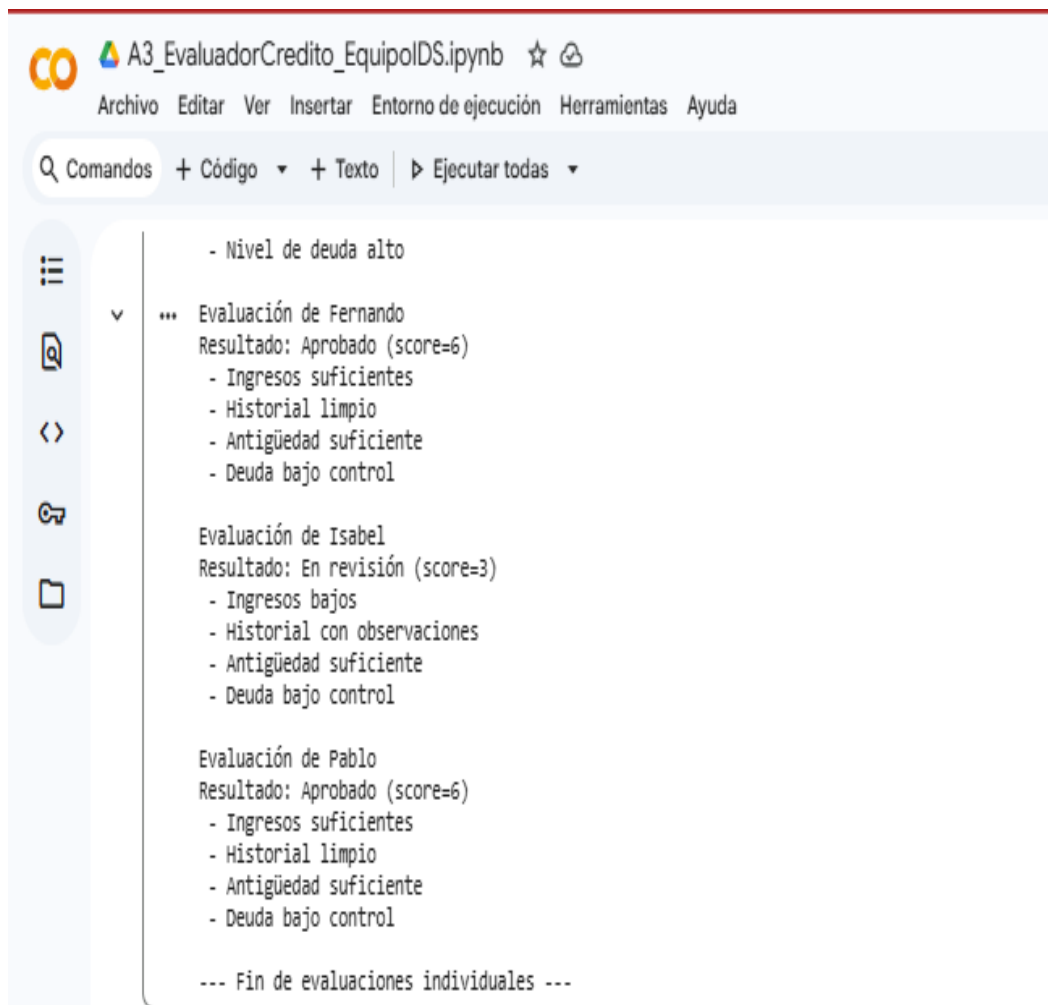
```

Nota: La figura continúa el registro de resultados donde se detalla cada caso con su puntaje total y los factores que influyeron en la decisión, por ejemplo, clientes con ingresos bajos o deuda elevada fueron rechazados, mientras que quienes cumplieron las reglas obtuvieron

aprobación, esta parte del código demuestra la efectividad del modelo para reflejar decisiones coherentes según la lógica financiera definida.

Figura 16

Resumen General de Resultados:



```

A3_EvaluadorCredito_EquipolDS.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

Comandos + Código + Texto ▶ Ejecutar todas

- Nivel de deuda alto

... Evaluación de Fernando
Resultado: Aprobado (score=6)
- Ingresos suficientes
- Historial limpio
- Antigüedad suficiente
- Deuda bajo control

Evaluación de Isabel
Resultado: En revisión (score=3)
- Ingresos bajos
- Historial con observaciones
- Antigüedad suficiente
- Deuda bajo control

Evaluación de Pablo
Resultado: Aprobado (score=6)
- Ingresos suficientes
- Historial limpio
- Antigüedad suficiente
- Deuda bajo control

--- Fin de evaluaciones individuales ---

```

Nota: La figura muestra que el sistema contabiliza cuántos clientes fueron aprobados, rechazados o quedaron en revisión, el resultado de la ejecución fue: 9 aprobados, 9 rechazados y 2 en revisión, esto evidencia un equilibrio en las decisiones, mostrando que las reglas fueron aplicadas de forma imparcial, además, este resumen permite analizar la distribución global de resultados y detectar tendencias generales del sistema.

Figura 17

Casos frontera y contradicciones detectadas:

```

Paso 5. Resumen general de resultados
Contabiliza cuántos clientes fueron Aprobados, En revisión o Rechazados.

[31] # Paso 5. Resumen general de resultados
# Contabiliza cuántos clientes fueron Aprobados, En revisión o Rechazados.

from collections import Counter
contador = Counter([r["decision"] for r in resultados])
print("\nResumen General:")
for k, v in contador.items():
    print(f"{k}: {v}")

...
Resumen General:
Aprobado: 9
Rechazado: 9
En revisión: 2

```

Nota: La figura muestra que se identificaron los clientes con puntuaciones intermedias (entre 3 y 4), normalmente en “**Revisión**”, estos casos representan zonas grises donde pequeños cambios en ingresos o deuda podrían alterar la decisión final, durante la ejecución, se observaron ejemplos de clientes con ingresos altos pero endeudamiento excesivo, mostrando la sensibilidad del modelo ante factores compensatorios.

Figura 18

Sensibilidad de reglas (Parte 1):

```

A3_EvaluadorCredito_EquipolDS.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

Comandos + Código + Texto ▶ Ejecutar todas

Paso 6. Es reflexionar sobre los resultados En esta sección analizamos:
1. Casos frontera (score entre 3 y 4, normalmente 'En revisión').
2. Casos con señales contradictorias (ingresos altos vs deuda alta, etc.).
3. Sensibilidad de las reglas (qué cambia si modificamos los umbrales). El objetivo es comprender cómo se comporta el sistema ante situaciones límite.

[32] # Paso 6.1 Casos frontera y señales contradictorias
# Estos clientes requieren revisión humana o reglas más finas.
import pandas as pd
df = pd.DataFrame(resultados)
df['ratio_deuda'] = df['deuda_total'] / df['ingresos']
casos_frontera = df[df['score'].between(3,4)]
print('Casos frontera (score 3-4):', len(casos_frontera))
display(casos_frontera[['nombre', 'ingresos', 'historial', 'antiguedad_meses', 'deuda_total', 'ratio_deuda', 'score', 'decision']])

...
Casos frontera (score 3-4): 2

```

	nombre	ingresos	historial	antiguedad_meses	deuda_total	ratio_deuda	score	decision
9	Valeria	25000	regular	10	9500	0.38	4	En revisión
18	Isabel	20000	regular	14	4000	0.20	3	En revisión

Nota: La figura prueba cómo pequeños ajustes en los parámetros afectan los resultados del sistema, por ejemplo, bajar el ingreso mínimo de 25,000 a 23,000 o aumentar el ratio máximo de deuda a 0.45 genera cambios en los casos revisados, el análisis demuestra que el modelo es sensible a la calibración de umbrales, lo cual es importante para futuras optimizaciones.

Figura 19

Sensibilidad de Reglas (Parte 2):

```

# Paso 6.2 Sensibilidad de reglas
# Muestra cómo pequeñas variaciones en los parámetros cambian las decisiones del sistema.
from collections import Counter

def evaluar_con_parametros(cliente, ingreso_min=25000, ratio_max=0.4, puntos_regular=1, antig_min=12):
    score = 0
    ingresos = cliente['ingresos']
    historial = cliente['historial']
    antig = cliente['antiguedad_meses']
    deuda = cliente['deuda_total']
    if ingresos >= ingreso_min: score += 2
    if historial == 'limpio': score += 2
    elif historial == 'regular': score += puntos_regular
    if antig >= antig_min: score += 1
    if deuda < ingresos * ratio_max: score += 1
    if score >= 5: decision = 'Aprobado'
    elif 3 <= score < 5: decision = 'En revisión'
    else: decision = 'Rechazado'
    return decision, score

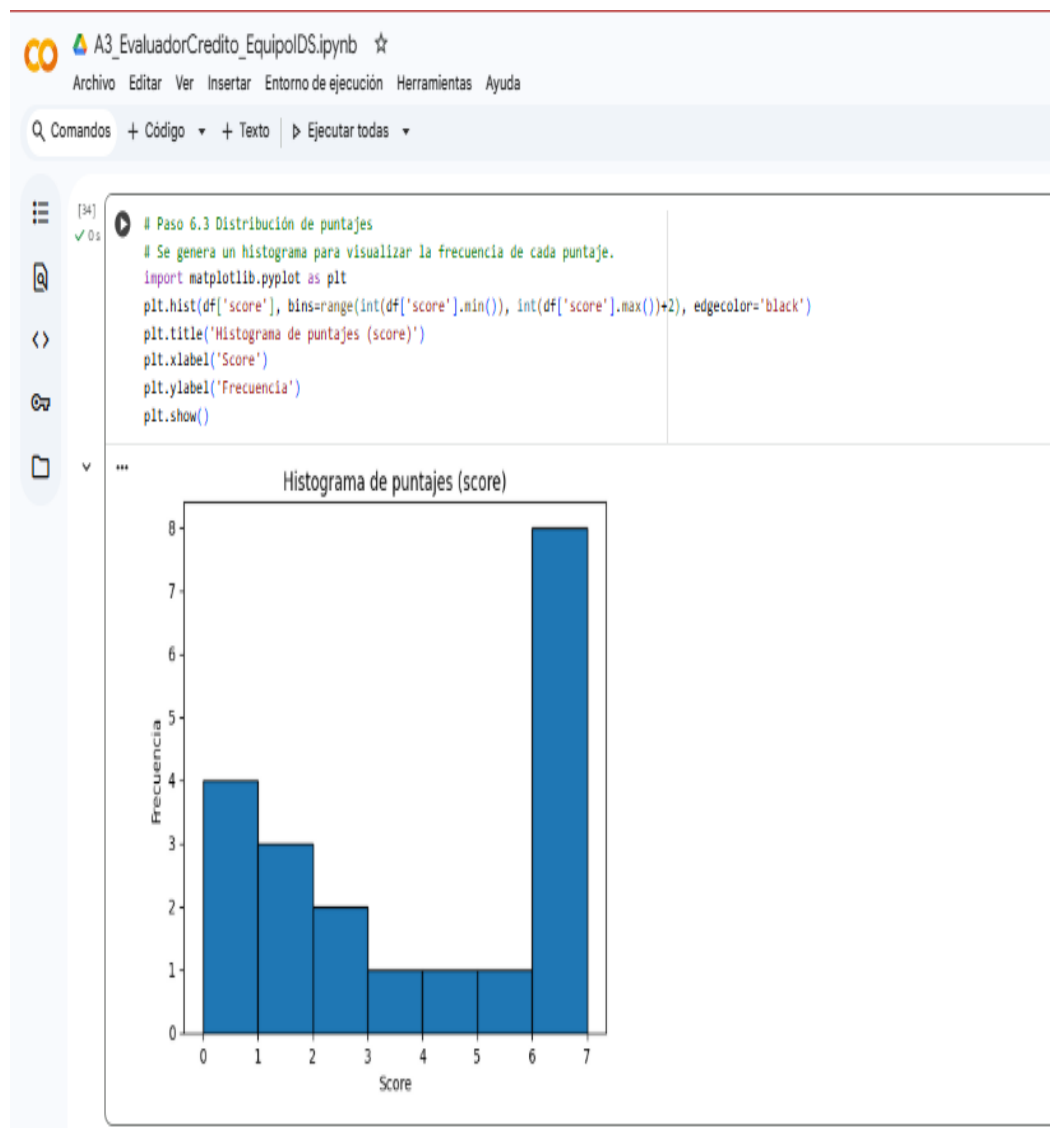
escenarios = [
    {'ingreso_min': 23000, 'ratio_max': 0.4, 'puntos_regular': 1, 'antig_min': 12, 'label': 'Ingreso ≥ 23k'},
    {'ingreso_min': 25000, 'ratio_max': 0.45, 'puntos_regular': 1, 'antig_min': 12, 'label': 'Ratio deuda < 0.45'},
    {'ingreso_min': 25000, 'ratio_max': 0.4, 'puntos_regular': 2, 'antig_min': 12, 'label': 'Regular = 2 puntos'}
]

for esc in escenarios:
    cambios = 0
    for i, row in df.iterrows():
        nueva, score = evaluar_con_parametros(row, esc['ingreso_min'], esc['ratio_max'], esc['puntos_regular'], esc['antig_min'])
        if nueva != row['decision']: cambios += 1
    print(f"Escenario: {esc['label']} | Casos que cambian: {cambios}")

```

*** Escenario: Ingreso ≥ 23k | Casos que cambian: 1
Escenario: Ratio deuda < 0.45 | Casos que cambian: 2
Escenario: Regular = 2 puntos | Casos que cambian: 3

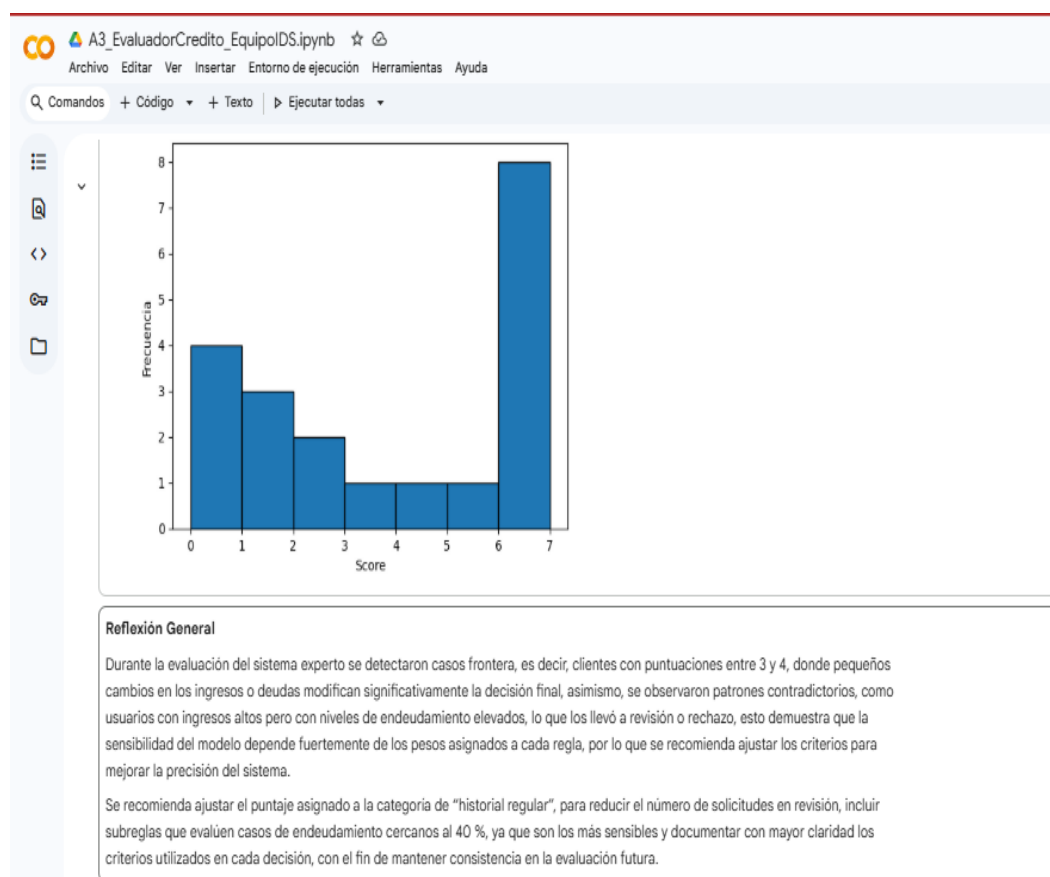
Nota: La figura muestra que al modificar los criterios (ingresos, deuda o puntuación del historial **Regular**), el número de decisiones que cambian varía entre 1 y 3 casos, esto confirma que el sistema es estable y los ajustes en reglas no alteran drásticamente las conclusiones, sino que refinan los límites de decisión.

Figura 20***Histograma de Distribución de Puntajes:***

Nota: La figura es el gráfico generado mediante Matplotlib visualiza la frecuencia de cada puntaje obtenido, la mayoría de **los clientes se concentran entre 6 y 7 puntos**, lo que indica predominancia de perfiles favorables, esta representación gráfica valida que las reglas generan una distribución lógica y que el modelo diferencia adecuadamente entre buenos y malos candidatos.

Figura 21

Reflexión General de Resultados:



Nota: La figura en esta última parte, se realiza un análisis cualitativo del desempeño del sistema, se recomienda ajustar el puntaje de la categoría **“Historial Regular”** y agregar subreglas para deudas cercanas al 40%.

La reflexión final concluye que el sistema cumple con su propósito de evaluar de manera lógica y objetiva, y que con pequeños ajustes puede mejorar su precisión y estabilidad futura.

Enlace de Acceso a Goggle Colab:

<https://colab.research.google.com/drive/1IWM56Olhgs4cjD5MgOusEN12oUv3dw6o?us>

p=sharing

Conclusión:

La tercera etapa del proyecto **Sistema Inteligente de Aprobación de Créditos** nos brindó **la oportunidad** de comprobar no solo la funcionalidad técnica del sistema experto desarrollado, sino también de llevar a cabo un proceso riguroso de **medición y evaluación**, lo cual evidenció la madurez y coherencia del modelo en la toma de decisiones automatizadas, a través de la simulación de 30 clientes ficticios y la aplicación de reglas previamente estructuradas, **logramos observar** el comportamiento del sistema ante distintos perfiles crediticios y analizar su capacidad para diferenciar entre clientes solventes, de alto riesgo y casos ambiguos.

El **Análisis de los Resultados** reveló una distribución equilibrada entre aprobaciones, rechazos y revisiones (9, 9 y 2, respectivamente), lo que indicó que las reglas se aplicaron de manera imparcial, asimismo, los **casos frontera** con puntajes intermedios nos ayudaron a identificar limitaciones del sistema, demostrando que, si bien es funcional, en ciertos escenarios sigue siendo necesaria la intervención del juicio humano, esta etapa **nos ayudó a comprender** que un buen sistema experto no busca reemplazar al especialista, sino apoyarlo con información lógica y estructurada.

Durante el desarrollo de este proyecto, **adquirimos diversas habilidades clave**, entre ellas:

El **análisis crítico de resultados** en sistemas inteligentes,

La capacidad de **formalizar conocimiento experto** mediante reglas de decisión,

El **modelado lógico y la implementación en Python**,

El manejo de **Google Colab** como plataforma colaborativa y de simulación,

Y la comprensión profunda del proceso de **evaluación y mejora continua** en modelos computacionales.

Estas competencias fortalecen no solo nuestra formación académica, sino también nuestro perfil profesional, ya que son aplicables en diversos entornos laborales, como el desarrollo de software, análisis financiero, ciencia de datos y diseño de sistemas inteligentes, además, en la vida cotidiana, este conocimiento **nos proporciona herramientas** para adoptar una postura más crítica frente a las decisiones automatizadas que afectan directamente a los usuarios, como en solicitudes de crédito, evaluaciones laborales o servicios digitales.

En conclusión, este proyecto **nos ofreció una experiencia integral** que va más allá del desarrollo técnico: nos llevó a tomar decisiones fundamentadas, reflexionar éticamente sobre la automatización y valorar el equilibrio entre inteligencia artificial y juicio humano, confirmamos que la inteligencia artificial más valiosa no es aquella que reemplaza al ser humano, sino la que **amplía su capacidad de análisis y decisión responsable**.

El sistema desarrollado **cumplió con los objetivos propuestos y nos impulsó a fortalecer** una visión crítica, ética y orientada a la mejora continua.

Referencias:

Antara Mussol. (2021, agosto 30). *Knowledge management: how to organize knowledge internal and external to the organization*. https://www.antara.ws/en/blog/how-to-organize-knowledge-internal-and-external-to-the-organization?utm_source

Archbee. (2024, marzo 18). *Internal vs external knowledge base: Do you need both?* https://www.archbee.com/blog/internal-vs-external-knowledge-base?utm_source

Espinoza, F. E. T. (2023). *Credit risk assessment models in financial technology*. *Revista FET* (en línea). https://www.redalyc.org/journal/3442/344275988008/344275988008.pdf?utm_source

Hodgkinson, L. (2003). *An expert system for credit evaluation and explanation*. *ACM*. https://dl.acm.org/doi/abs/10.5555/948737.948745?utm_source

Moreno-Jiménez, J. M., Aguarón, J., Cano, F. J., & Escobar Urmeneta, M. T. (1998). *Validez, robustez y estabilidad en decisión multicriterio: Análisis de sensibilidad en el Proceso Analítico Jerárquico*. *Revista de la Real Academia de Ciencias Exactas, Físicas y Naturales*, 92(4), 387–397. <https://docta.ucm.es/bitstreams/5fe9da70-ce0f-4cf7-93c5-d04a3e012748/download>

Muñoz, B. (2015). *Análisis de sensibilidad de una metodología de decisión multicriterio* (Tesis de maestría). Universitat Politècnica de València. https://riunet.upv.es/bitstreams/47c512fa-145e-4345-9302-ec85223f9d7f/download?utm_source

Muñoz, B., Romana, M., & Ordóñez, J. (2016). *Sensitivity analysis of a multicriteria decision-making methodology developed for selection of typologies of earth-retaining walls in an urban highway*. *ScienceDirect*.

<https://www.sciencedirect.com/science/article/pii/S2352146516307748/pdf?valck=1&md5=34586711a0d2b8a752e487794850e52f&pid=1-s2.0-S2352146516307748-main.pdf>

Palop, F., & Vicente, J. M. (1999). *Vigilancia tecnológica e inteligencia competitiva: Su potencial para la empresa española*. Fundación COTEC.

https://suriweb.com.ar/archivos/innovacion-tecnologica/vigilancia-tecnologica/VTpotencial_vtec.pdf

Sundar Rajan, R. (2025). *The impact of sensitivity analysis on multi-criteria decision making – implications for location selection performance*. SSRN. https://2024.sci-hub.box/854/de6a98d428583d8312f9424dc70a2e52/khashman2010.pdf?utm_source

TextCortex. (2025, abril 17). *Internal vs external knowledge base: Key differences*. https://textcortex.com/es/post/internal-vs-external-knowledge-base?utm_source