GUIA COMPLETA DE INVESTIGACION

Para Primera Tesis de Ingenieria de Sistemas

Sistema Web de Analisis Financiero con Machine Learning para PYMES de Ibague, Tolima

Esta guia esta disenada especificamente para estudiantes que realizan su primera tesis de grado. Contiene instrucciones paso a paso, recursos recomendados, consejos practicos y listas de verificacion para cada fase del proyecto.

RECURSOS CONFIRMADOS PARA ESTE PROYECTO:

Datos: SIREM (~9 GB) + Camara de Comercio Ibague (disponibles y gratuitos)
Hardware: Intel i5-10600KF, 16GB RAM, RTX 2060 (suficiente)
Herramientas: VS Code, Python, Node.js, PostgreSQL (todas gratuitas)
Estrategia: JOIN por NIT para obtener PYMES de Ibague con datos financieros

Version 1.0 - February 2026

CONTENIDO DE LA GUIA

PARTE 1: FUNDAMENTOS DE INVESTIGACION

1.1 ¿Que es una tesis y que se espera de ti?

Una tesis de pregrado en Ingenieria de Sistemas es un proyecto que demuestra tu capacidad para resolver un problema real aplicando los conocimientos adquiridos durante la carrera. **No se espera que inventes algo completamente nuevo**, sino que apliques metodologias y tecnologias existentes de manera rigurosa y documentada.

Lo que los jurados evaluaran:

- **Planteamiento claro:** ¿Entiendes el problema que quieres resolver?
- **Justificacion solida:** ¿Por que es importante este proyecto?
- **Metodologia apropiada:** ¿Seguiste un proceso ordenado y documentado?
- **Implementacion funcional:** ¿El sistema hace lo que prometiste?
- **Resultados medibles:** ¿Puedes demostrar que funciona?
- **Documentacion completa:** ¿El documento esta bien escrito y estructurado?
- **Defensa oral:** ¿Puedes explicar y defender tu trabajo?

1.2 Estructura del documento de grado

Seccion	Contenido	Extension aprox.
Preliminares	Portada, dedicatoria, agradecimientos, tabla de contenido, listas	5-10 paginas
Introduccion	Contexto, problema, objetivos, alcance	3-5 paginas
Marco teorico	Conceptos, teorias, estado del arte	15-25 paginas
Metodologia	Tipo de investigacion, fases, herramientas	5-10 paginas
Desarrollo	Requerimientos, diseno, implementacion	30-50 paginas
Resultados	Pruebas, metricas, analisis	10-15 paginas
Conclusiones	Logros, limitaciones, trabajo futuro	3-5 paginas
Referencias	Bibliografia en formato APA/IEEE	3-5 paginas
Anexos	Codigo, manuales, encuestas	Variable

1.3 Como organizar tu tiempo

Una tesis de 16 semanas requiere dedicacion constante. Aqui hay un plan realista:

Semanas	Horas/semana	Actividad principal	Entregable

1-2	15-20h	Investigacion + datos	Marco teorico inicial
3-4	15-20h	Diseno del sistema	Diagramas UML
5-8	20-25h	Desarrollo core	Modulos principales
9-12	20-25h	ML + Dashboard	Sistema funcional
13-14	15-20h	Pruebas	Resultados documentados
15-16	25-30h	Documento + sustentacion	Tesis completa

CONSEJO IMPORTANTE: No dejes la documentacion para el final. Escribe el marco teorico mientras investigas, documenta el desarrollo mientras programas, y anota los resultados mientras pruebas. Esto te ahorrara semanas de estres.

PARTE 2: INVESTIGACION BIBLIOGRAFICA

2.1 Temas que debes investigar

Tu marco teorico debe cubrir los siguientes temas. Para cada uno, necesitas definiciones, explicaciones y referencias academicas:

Tema	Subtemas a cubrir	Referencias sugeridas
Estados financieros bajo NIIF	- Balance General - Estado de Resultados - Flujo de Efectivo - NIIF para PYMES (Grupo 2)	IASB (2015), Decreto 3022, Ortiz Anaya (2018)
Indicadores financieros	- Liquidez - Rentabilidad - Endeudamiento - Eficiencia	Gitman & Zutter (2016), Ross et al. (2019)
PYMES en Colombia	- Definicion legal - Clasificacion por tamano - Problematica financiera - Estadisticas de mortalidad	Ley 590/2000, Ley 905/2004, Confecamaras, DANE
Sistemas de informacion	- Definicion de SI - Arquitectura web - Bases de datos - APIs REST	Laudon & Laudon (2020), Pressman (2015)
Machine Learning	- Aprendizaje supervisado - Clasificacion - Random Forest, XGBoost - Metricas de evaluacion	Hastie et al. (2009), Chen & Guestrin (2016)
Evaluacion de riesgo financiero	- Modelos de scoring - Altman Z-Score - Indicadores de alerta - Prediccion de quiebra	Altman (1968), Ohlson (1980)

2.2 Donde buscar informacion academica

Fuente	URL	Tipo de contenido
Google Scholar	scholar.google.com	Articulos academicos, tesis, libros
Scielo	scielo.org	Revistas cientificas latinoamericanas
Redalyc	redalyc.org	Revistas de acceso abierto
Dialnet	dialnet.unirioja.es	Articulos en espanol

IEEE Xplore	ieeexplore.ieee.org	Papers de ingenieria y computacion
ResearchGate	researchgate.net	Papers y contacto con autores
Repositorios universitarios	repositorio.[universidad].edu.co	Tesis de grado similares
Biblioteca universidad	Fisica o virtual	Libros de texto, bases de datos

TIPS DE BUSQUEDA:
- Usa terminos en ingles para encontrar mas resultados: "financial ratio analysis", "machine learning credit risk", "SME financial distress prediction"
- Filtra por ano (ultimos 5-10 anos) para informacion actualizada
- Busca tesis similares en repositorios de universidades colombianas

2.3 Como gestionar referencias

Usa un gestor bibliografico desde el inicio. Te ahorrara horas al momento de citar y generar la bibliografia:

Herramienta	Costo	Ventajas
Zotero	Gratis	Plugin para navegador, sincroniza con Word/Google Docs
Mendeley	Gratis	Buen lector de PDF, red social academica
EndNote	Pago (gratis via universidad)	Muy completo, estandar en muchas universidades

FORMATO DE CITACION: Verifica con tu universidad si usan APA, IEEE u otro formato. Para Ingenieria de Sistemas, IEEE es comun. Ejemplo IEEE:

```
[1] L. J. Gitman and C. J. Zutter, "Principios de administracion financiera," 14th ed. Mexico: Pearson, 2016.
```

PARTE 3: OBTENCION Y PROCESAMIENTO DE DATOS

3.1 Descarga de datasets

Necesitas descargar dos conjuntos de datos principales:

Dataset 1: SIREM (Supersociedades)

```
Fuente: https://www.datos.gov.co (buscar "Estados Financieros NIIF")

Archivos a descargar:
- Estados_Financieros_NIIF-_Caratula.csv (2.07 GB)
- Estados_Financieros_NIIF-_Estado_de_Situacion_Financiera.csv (4.10 GB)
- Estados_Financieros_NIIF-_Estado_de_Resultado_Integral.csv (1.57 GB)
- Estados_Financieros_NIIF-_Estado_de_Flujo_Efectivo.csv (1.43 GB)

Total: ~9.17 GB
```

Dataset 2: Camara de Comercio de Ibague

```
Fuente: https://www.datos.gov.co/Comercio-Industria-y-Turismo/
BASE-DE-DATOS-DE-EMPRESAS-Y-O-ENTIDADES-ACTIVAS-JU/gwqv-sqvs

Campos disponibles:
- NIT (clave para el JOIN)
- Razon social
- Actividad economica (CIIU)
- Tamano empresarial
- Municipio (para confirmar que es Ibague)
```

3.2 Integracion de datos (JOIN)

El objetivo es obtener solo las empresas de Ibague que tienen datos financieros en SIREM. Aqui esta el proceso paso a paso:

```
import pandas as pd

# PASO 1: Cargar empresas de Ibague (archivo pequeno)
camara_ibague = pd.read_csv('empresas_camara_comercio_ibague.csv')
nits_ibague = set(camara_ibague['NIT'].astype(str).str.replace('[^0-9]', '', regex=True))
print(f"Empresas en Ibague: {len(nits_ibague)}")

# PASO 2: Filtrar SIREM por PYMES y por NITs de Ibague (usar chunks por tamano)
def procesar_sirem(archivo, nits_ibague):
chunks = pd.read_csv(archivo, chunksize=100000, low_memory=False)
resultado = []
for chunk in chunks:
# Filtrar PYMES
chunk = chunk[chunk['PUNTO_ENTRADA'] == 'NIIF Pymes']
# Limpiar NIT y filtrar por Ibague
chunk['NIT_CLEAN'] = chunk['NIT'].astype(str).str.replace('[^0-9]', '', regex=True)
chunk = chunk[chunk['NIT_CLEAN'].isin(nits_ibague)]
resultado.append(chunk)
return pd.concat(resultado, ignore_index=True)

# PASO 3: Procesar cada archivo
```

```
balance_ibague = procesar_sirem('Estado_Situacion_Financiera.csv', nits_ibague)
resultados_ibague = procesar_sirem('Estado_Resultado_Integral.csv', nits_ibague)
flujo_ibague = procesar_sirem('Estado_Flujo_Efectivo.csv', nits_ibague)

# PASO 4: Guardar datasets filtrados
balance_ibague.to_csv('balance_pymes_ibague.csv', index=False)
print(f"Empresas PYMES de Ibague con datos financieros: {balance_ibague['NIT'].nunique()}")
```

3.3 Limpieza y preparacion

- **Normalizar NITs:** Eliminar puntos, comas, guiones. Convertir a string.
- **Manejar valores nulos:** Decidir si eliminar o imputar (promedio, mediana).
- **Verificar tipos de datos:** VALOR debe ser numerico, FECHA debe ser datetime.
- **Eliminar duplicados:** Mismo NIT + CONCEPTO + PERIODO no debe repetirse.
- **Pivotear datos:** Convertir de formato vertical (concepto por fila) a horizontal (concepto por columna).

PARTE 4: DESARROLLO DEL SISTEMA

4.1 Configuracion del entorno

Herramienta	Version	Instalacion
Node.js	v18 LTS o superior	nodejs.org
Python	3.10+	python.org
PostgreSQL	15+	postgresql.org
VS Code	Ultima	code.visualstudio.com
Git	Ultima	git-scm.com
DBeaver	Ultima	dbeaver.io (cliente BD)

4.2 Estructura del proyecto

```
proyecto-tesis/
|-- frontend/ # Aplicacion React
| |-- src/
| | |-- components/ # Componentes reutilizables
| | |-- pages/ # Paginas principales
| | |-- services/ # Llamadas a API
| | |-- hooks/ # Custom hooks
| | |-- utils/ # Funciones auxiliares
| |-- package.json
|
|-- backend/ # API Node.js
| |-- src/
| | |-- controllers/ # Logica de endpoints
| | |-- models/ # Modelos Prisma
| | |-- routes/ # Definicion de rutas
| | |-- middleware/ # Autenticacion, validacion
| | |-- services/ # Logica de negocio
| | |-- utils/ # Calculos financieros
| |-- prisma/schema.prisma # Esquema de BD
| |-- package.json
|
|-- ml-service/ # Servicio de ML en Python
| |-- src/
| | |-- train.py # Entrenamiento del modelo
| | |-- predict.py # API de prediccion
| | |-- preprocessing.py # Preparacion de datos
| |-- models/ # Modelos entrenados (.pkl)
| |-- requirements.txt
|
|-- data/ # Datos procesados
|-- docs/ # Documentacion
|-- docker-compose.yml # Configuracion de contenedores
|-- README.md
```

4.3 Desarrollo por modulos (orden sugerido)

Orden	Modulo	Tecnologias	Tiempo est.
1	Base de datos	PostgreSQL + Prisma	3-4 dias
2	Autenticacion	JWT + bcrypt	4-5 dias
3	CRUD Empresas	Express + React	5-6 dias
4	Estados Financieros	Forms + Validacion	7-10 dias
5	Motor de Indicadores	JavaScript/Python	5-7 dias
6	Modelo ML	Scikit-learn + XGBoost	7-10 dias
7	Dashboard	Recharts + React	5-7 dias
8	Reportes PDF	jsPDF	3-4 dias

PARTE 5: MACHINE LEARNING

5.1 Preparacion de datos para ML

El modelo de ML necesita datos en formato horizontal con indicadores calculados y una variable objetivo (target). Aqui esta el proceso:

```
# PASO 1: Calcular indicadores por empresa y ano
def calcular_indicadores(balance, resultados):
indicadores = pd.DataFrame()
indicadores['NIT'] = balance['NIT'].unique()

# Ejemplo: Razon Corriente
activo_corriente = balance[balance['CONCEPTO'].str.contains('Activo.*corriente', case=False)]
pasivo_corriente = balance[balance['CONCEPTO'].str.contains('Pasivo.*corriente', case=False)]
indicadores['razon_corriente'] = activo_corriente['VALOR'] / pasivo_corriente['VALOR']

# Repetir para cada indicador...
return indicadores

# PASO 2: Crear variable objetivo (etiqueta de riesgo)
def crear_etiqueta_riesgo(row):
if (row['razon_corriente'] < 1.0 and
row['endeudamiento'] > 0.7 and
row['margen_neto'] < 0):
return 2 # ALTO RIESGO
elif (row['razon_corriente'] < 1.5 or
row['endeudamiento'] > 0.5 or
row['margen_neto'] < 0.05):
return 1 # RIESGO MEDIO
else:
return 0 # BAJO RIESGO

indicadores['riesgo'] = indicadores.apply(crear_etiqueta_riesgo, axis=1)
```

5.2 Entrenamiento del modelo

```
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import classification_report, roc_auc_score
import joblib

# Separar features y target
X = indicadores.drop(['NIT', 'riesgo'], axis=1)
y = indicadores['riesgo']

# Split train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Entrenar modelo XGBoost
modelo = XGBClassifier(n_estimators=100, max_depth=5, random_state=42)
modelo.fit(X_train, y_train)

# Evaluar
y_pred = modelo.predict(X_test)
```

```
print(classification_report(y_test, y_pred))
print(f"AUC-ROC: {roc_auc_score(y_test, modelo.predict_proba(X_test), multi_class='ovr')}")

# Guardar modelo
joblib.dump(modelo, 'modelo_riesgo_pymes.pkl')
```

5.3 Metricas objetivo

Metrica	Objetivo minimo	Descripcion
Accuracy	>= 70%	Porcentaje de predicciones correctas
Precision	>= 70%	De los que predigo riesgo, cuantos lo son
Recall	>= 65%	De los que son riesgo, cuantos detecto
F1-Score	>= 67%	Balance entre precision y recall
AUC-ROC	>= 75%	Capacidad discriminativa del modelo

PARTE 6: DOCUMENTACION Y SUSTENTACION

6.1 Consejos para la redaccion

- **Escribe en tercera persona:** 'Se desarrolló...' en lugar de 'Yo desarrollé...'
- **Usa tiempo pasado para resultados:** 'El sistema logró una precision de 78%'
- **Cita todo:** Cada afirmación importante necesita una referencia [1]
- **Numera figuras y tablas:** 'Como se observa en la Figura 3.2...'
- **Evita coloquialismos:** No uses 'o sea', 'básicamente', 'etc.'
- **Revisa ortografía:** Usa correctores como LanguageTool (gratis)
- **Pide revisión:** Que alguien más lea tu documento antes de entregar

6.2 Preparacion de la sustentacion

La sustentación típicamente dura 30-45 minutos: 15-20 de presentación y 15-20 de preguntas. Aquí está la estructura sugerida:

Seccion	Tiempo	Contenido
Introduccion	2 min	Saludo, titulo, agenda
Problema	2 min	Contexto, problema, pregunta
Objetivos	1 min	General y especificos
Marco teorico	3 min	Conceptos clave (resumido)
Metodologia	2 min	Fases, tecnologias
Desarrollo	5 min	Arquitectura, modulos principales
Demo	3-5 min	Demostracion en vivo del sistema
Resultados	3 min	Metricas, pruebas, cumplimiento
Conclusiones	2 min	Logros, limitaciones, trabajo futuro

Preguntas frecuentes de los jurados:

- ¿Por qué eligió esta metodología y no otra?
- ¿Cuál es el aporte innovador de su trabajo?
- ¿Cómo garantiza la seguridad de los datos financieros?
- ¿Qué tan preciso es el modelo de ML? ¿Cómo lo validó?
- ¿Qué pasaría si una empresa ingresa datos incorrectos?
- ¿Cómo se compara su solución con las existentes?
- ¿Cuáles son las limitaciones de su sistema?

- ¿Cómo escalaría la solución a nivel nacional?

ANEXOS: CHECKLISTS Y RECURSOS

Checklist General del Proyecto

```
[ ] Propuesta aprobada por el director
[ ] Marco teorico con 25+ referencias
[ ] Datos SIREM descargados
[ ] Datos Camara de Comercio descargados
[ ] JOIN realizado - PYMES de Ibague identificadas
[ ] Datos limpios y procesados
[ ] Indicadores financieros calculados
[ ] Modelo de ML entrenado y evaluado
[ ] Sistema web funcional
[ ] Pruebas documentadas
[ ] Documento de grado completo
[ ] Presentacion preparada
[ ] Demo funcionando
```

Recursos de Aprendizaje Recomendados

Tema	Recurso	Idioma
React.js	Curso de Midudev (YouTube)	Espanol
Node.js	Curso de Fazt Code (YouTube)	Espanol
PostgreSQL	Tutorial oficial postgresql.org	Ingles
Machine Learning	Curso de Andrew Ng (Coursera)	Ingles (subtitulos)
XGBoost	Documentacion oficial xgboost.readthedocs.io	Ingles
Git/GitHub	Curso de HolaMundo (YouTube)	Espanol
Analisis Financiero	Libro Ortiz Anaya - Biblioteca	Espanol

RECUERDA: Una tesis no tiene que ser perfecta, tiene que estar terminada. Es mejor entregar algo funcional y bien documentado que perseguir la perfeccion y no terminar. Tu director de tesis esta para guiarte - no dudes en pedir ayuda cuando la necesites. ¡Exitos en tu trabajo de grado!