

INFORME DE EVALUACIÓN DE PROYECTO

Sistema de Predicción Geoespacial de Tarifas Eléctricas en Chile

Curso de Geoinformática - 2° Semestre 2025
Universidad de Santiago de Chile

Profesor: Dr. Francisco Parra O.
Tipo de Proyecto: Comercial

Agosto 2025

Índice

1. Resumen Ejecutivo	3
1.1. Evaluación General	3
2. Análisis Técnico Detallado	3
2.1. Arquitectura del Sistema	3
2.1.1. Capa de Datos	3
2.1.2. Capa de Procesamiento	4
2.1.3. Capa de Visualización	4
2.2. Variables Clave y Justificación	5
2.3. Metodología de Machine Learning	5
2.3.1. Preprocesamiento Espacial	5
2.3.2. Modelo Predictivo	5
3. Evaluación de Complejidad y Alcance	7
3.1. Aspectos Destacables	7
3.2. Matriz de Riesgos y Mitigación	7
4. Plan de Implementación Recomendado	8
4.1. Fase 1: Preparación de Datos (3 semanas)	8
4.2. Fase 2: Desarrollo del Modelo (4 semanas)	8
4.3. Fase 3: Sistema de Visualización (3 semanas)	8
4.4. Fase 4: Validación y Ajustes (2 semanas)	8
5. Mejoras y Extensiones Sugeridas	8
5.1. Mejoras Técnicas Inmediatas	8
5.2. Extensiones Comerciales	9
6. Impacto Esperado	10
6.1. Beneficiarios Directos	10
6.2. Métricas de Éxito	10
7. Conclusiones y Recomendaciones	10
7.1. Fortalezas del Proyecto	10
7.2. Recomendaciones Finales	11
7.3. Siguiendo Pasos	11
8. Referencias Técnicas Sugeridas	11

1. Resumen Ejecutivo

El proyecto propuesto representa una **excelente integración** de tecnologías geoespaciales avanzadas para resolver un problema crítico del mercado energético chileno. La propuesta combina:

- **Google Earth Engine** para obtención de variables climáticas
- **Machine Learning** con algoritmos de series temporales
- **Análisis espacial** del sistema eléctrico nacional
- **Visualización interactiva** de predicciones georreferenciadas

1.1. Evaluación General

Criterio	Puntuación	Observación
Relevancia técnica	9.5/10	Uso avanzado de GEE y ML espacial
Viabilidad	9.0/10	Datos disponibles públicamente
Innovación	8.5/10	Enfoque novedoso para mercado chileno
Impacto comercial	9.0/10	Alto valor para múltiples stakeholders
Complejidad apropiada	9.5/10	Desafiante para 8° semestre
TOTAL	91/100	PROYECTO SOBRESALIENTE

2. Análisis Técnico Detallado

2.1. Arquitectura del Sistema

El proyecto propone una arquitectura de tres capas bien definida:

2.1.1. Capa de Datos

- **Datos Satelitales (Google Earth Engine)**
 - MODIS para temperatura superficial (LST)
 - ERA5 para radiación solar
 - Sentinel-2 para índices espectrales (NDWI, NDSI)
- **Datos del Mercado Eléctrico**
 - Costos marginales históricos (Coordinador Eléctrico Nacional)
 - Precios de combustibles fósiles
 - Capacidad instalada por tecnología

2.1.2. Capa de Procesamiento

```
1 # 1. Extraccion de variables GEE
2 import ee
3 ee.Initialize()
4
5 def extract_climate_variables(roi, date_range):
6     # Temperatura superficial
7     lst = ee.ImageCollection('MODIS/006/MOD11A1')\
8         .filterDate(date_range[0], date_range[1])\
9         .filterBounds(roi)\
10        .select('LST_Day_1km')\
11        .mean()
12
13    # Radiacion solar
14    solar = ee.ImageCollection('ECMWF/ERA5/DAILY')\
15        .filterDate(date_range[0], date_range[1])\
16        .select('surface_solar_radiation_downwards')\
17        .mean()
18
19    # Indices de sequia
20    ndwi = calculate_ndwi(roi, date_range)
21
22    return ee.Image.cat([lst, solar, ndwi])
23
24 # 2. Modelo predictivo
25 from sklearn.ensemble import RandomForestRegressor
26 import xgboost as xgb
27
28 def train_price_model(features, targets):
29     # Features: variables climaticas + historicos
30     # Targets: costos marginales por zona
31
32     model = xgb.XGBRegressor(
33         n_estimators=500,
34         max_depth=8,
35         learning_rate=0.01,
36         objective='reg:squarederror'
37     )
38
39     # Incluir lags temporales
40     features_with_lags = add_temporal_lags(features, [1, 7, 30,
41        365])
42
43     # Validacion cruzada espacial
44     cv_scores = spatial_cross_validation(model,
45        features_with_lags, targets)
46
47     return model.fit(features_with_lags, targets)
```

Listing 1: Pipeline de procesamiento propuesto

2.1.3. Capa de Visualización

- Dashboard interactivo con Streamlit/Dash
- Mapas de calor con predicciones usando Folium

- API REST para integración con sistemas externos

2.2. Variables Clave y Justificación

Variable	Fuente	Justificación	Resolución
NDWI (Índice de Agua)	Sentinel-2	Nivel de embalses afecta generación hidroeléctrica (40 % matriz Chile)	10m, quincenal
NDSI (Índice de Nieve)	MODIS/Sentinel-2	Predictor de deshielo y disponibilidad hídrica futura	500m, diaria
LST (Temperatura)	MODIS	Correlación con demanda por climatización	1km, diaria
Radiación Solar	ERA5	Predictor directo de generación solar (20 % matriz norte)	30km, horaria
Velocidad Viento	ERA5	Generación eólica en centro-sur	30km, horaria
Precipitación	GPM/CHIRPS	Afluentes futuros para hidroeléctricas	10km, diaria
CMg Histórico	CEN	Variable objetivo y feature lag	Por barra, horaria
Precio Combustibles	CNE	Costo generación térmica de respaldo	Nacional, semanal

2.3. Metodología de Machine Learning

2.3.1. Preprocesamiento Espacial

1. **Zonificación del territorio:** División en 5 macrozonas eléctricas
2. **Agregación temporal:** Promedios mensuales para predicción estratégica
3. **Normalización:** StandardScaler para variables climáticas
4. **Feature engineering:**
 - Ratios inter-zonales de variables
 - Anomalías respecto a promedios históricos
 - Índices compuestos (ej: stress hídrico = $f(\text{NDWI}, \text{precipitación}, \text{temperatura})$)

2.3.2. Modelo Predictivo

```

1 import xgboost as xgb
2 from sklearn.model_selection import TimeSeriesSplit
3 from sklearn.metrics import mean_absolute_percentage_error
4 import geopandas as gpd
5
6 class SpatialElectricityPricePredictor:
7     def __init__(self, zones_shapefile):

```

```

8         self.zones = gpd.read_file(zones_shapefile)
9         self.models = {} # Un modelo por zona
10
11     def train(self, climate_data, market_data):
12         for zone_id in self.zones['id']:
13             # Filtrar datos por zona
14             zone_features = self.extract_zone_features(
15                 climate_data, zone_id
16             )
17
18             # Configurar modelo con hiperparametros optimizados
19             self.models[zone_id] = xgb.XGBRegressor(
20                 n_estimators=1000,
21                 max_depth=10,
22                 learning_rate=0.01,
23                 subsample=0.8,
24                 colsample_bytree=0.8,
25                 gamma=0.1,
26                 reg_alpha=0.1,
27                 reg_lambda=1.0,
28                 objective='reg:squarederror',
29                 eval_metric='mape'
30             )
31
32             # Validacion temporal
33             tscv = TimeSeriesSplit(n_splits=5)
34             cv_scores = []
35
36             for train_idx, val_idx in tscv.split(zone_features):
37                 X_train = zone_features.iloc[train_idx]
38                 y_train = market_data['cmg'].iloc[train_idx]
39                 X_val = zone_features.iloc[val_idx]
40                 y_val = market_data['cmg'].iloc[val_idx]
41
42                 self.models[zone_id].fit(
43                     X_train, y_train,
44                     eval_set=[(X_val, y_val)],
45                     early_stopping_rounds=50,
46                     verbose=False
47                 )
48
49                 predictions = self.models[zone_id].predict(
50                     X_val)
51                 mape = mean_absolute_percentage_error(y_val,
52                                                         predictions)
53                 cv_scores.append(mape)
54
55                 print(f"Zona {zone_id} - MAPE promedio: {np.mean(
56                     cv_scores):.2%}")
57
58     def predict_spatial(self, future_climate):
59         predictions = {}
60         for zone_id in self.zones['id']:
61             zone_features = self.extract_zone_features(
62                 future_climate, zone_id
63             )

```

```

61         predictions[zone_id] = self.models[zone_id].predict
           (zone_features)
62
63     return self.create_prediction_map(predictions)

```

Listing 2: Implementación del modelo XGBoost

3. Evaluación de Complejidad y Alcance

3.1. Aspectos Destacables

1. Integración Multidisciplinaria

- Teledetección satelital
- Ciencia de datos espaciales
- Economía energética
- Meteorología aplicada

2. Desafíos Técnicos Apropriados

- Manejo de big data satelital en GEE
- Sincronización de múltiples fuentes temporales
- Modelado de dependencias espaciales
- Optimización de hiperparámetros en ML

3. Escalabilidad

- Arquitectura cloud-ready
- Procesamiento paralelo por zonas
- API para integración empresarial

3.2. Matriz de Riesgos y Mitigación

Riesgo	Probabilidad	Impacto	Mitigación
Calidad datos GEE	Baja	Alto	Validación con estaciones meteorológicas in-situ
Overfitting modelo	Media	Medio	Cross-validation espacial y temporal rigurosa
Cambios regulatorios	Baja	Medio	Diseño modular para adaptar nuevas variables
Latencia predicciones	Media	Bajo	Cache de resultados y procesamiento asíncrono

4. Plan de Implementación Recomendado

4.1. Fase 1: Preparación de Datos (3 semanas)

- Configurar cuenta Google Earth Engine
- Desarrollar scripts de extracción de variables climáticas
- Obtener y limpiar datos históricos del CEN
- Crear base de datos espacial con PostGIS

4.2. Fase 2: Desarrollo del Modelo (4 semanas)

- Análisis exploratorio de correlaciones
- Feature engineering espacial y temporal
- Entrenamiento y optimización de modelos
- Validación con métricas espaciales (Moran's I)

4.3. Fase 3: Sistema de Visualización (3 semanas)

- Dashboard interactivo con Streamlit
- Mapas de predicción con Folium
- API REST con FastAPI
- Documentación técnica

4.4. Fase 4: Validación y Ajustes (2 semanas)

- Backtesting con datos históricos
- Análisis de sensibilidad
- Optimización de performance
- Presentación final

5. Mejoras y Extensiones Sugeridas

5.1. Mejoras Técnicas Inmediatas

1. Incorporar Autocorrelación Espacial

```
1 from pysal.lib import weights
2 from pysal.model import spreg
3
4 # Crear matriz de pesos espaciales
5 w = weights.Queen.from_dataframe(zones_gdf)
6
```



```

7 # Modelo con lag espacial
8 model = spreg.ML_Lag(
9     y, X, w,
10     name_y='cmg',
11     name_x=feature_names
12 )

```

2. Análisis de Incertidumbre

- Intervalos de confianza con bootstrap
- Propagación de error desde inputs satelitales
- Escenarios probabilísticos

3. Deep Learning Espacial

```

1 import torch
2 import torch_geometric
3
4 class SpatialGNN(torch.nn.Module):
5     """Graph Neural Network para prediccion
6     considerando conectividad de la red electrica"""
7
8     def __init__(self, num_features, hidden_dim):
9         super().__init__()
10         self.conv1 = GCNConv(num_features, hidden_dim)
11         self.conv2 = GCNConv(hidden_dim, hidden_dim)
12         self.fc = torch.nn.Linear(hidden_dim, 1)
13
14     def forward(self, x, edge_index):
15         x = self.conv1(x, edge_index).relu()
16         x = self.conv2(x, edge_index).relu()
17         return self.fc(x)

```

5.2. Extensiones Comerciales

1. Módulo de Optimización para Generadoras

- Recomendaciones de despacho óptimo
- Análisis what-if para mantenimientos
- Alertas de oportunidades de arbitraje

2. Integración con Trading Energético

- Señales de compra/venta en mercado spot
- Valorización de contratos PPA
- Gestión de riesgo de portafolio

3. Servicios para Consumidores Industriales

- Optimización de consumo por horario
- Evaluación de autogeneración
- Negociación de contratos

6. Impacto Esperado

6.1. Beneficiarios Directos

- **Generadoras eléctricas:** Optimización de estrategias de oferta
- **Comercializadoras:** Mejor gestión de riesgo de precio
- **Grandes consumidores:** Planificación de consumo y costos
- **Reguladores:** Monitoreo de comportamiento del mercado
- **Inversionistas:** Evaluación de proyectos renovables

6.2. Métricas de Éxito

Métrica	Objetivo	Método de Medición
MAPE predicción	¡15 %	Backtesting 2023-2024
Cobertura espacial	95 % territorio	Zonas con predicción
Latencia API	¡500ms	Monitoring en producción
Usuarios activos	¡50/mes	Google Analytics
ROI para clientes	¡20 %	Encuesta post-implementación

7. Conclusiones y Recomendaciones

7.1. Fortalezas del Proyecto

1. **Alta relevancia:** Aborda problema crítico del sector energético chileno
2. **Innovación técnica:** Integración novedosa de GEE + ML + mercado eléctrico
3. **Viabilidad demostrada:** Todos los datos necesarios son accesibles
4. **Escalabilidad comercial:** Múltiples modelos de negocio viables
5. **Aprendizaje integral:** Cubre todos los objetivos del curso de Geoinformática

7.2. Recomendaciones Finales

RECOMENDACIÓN DEL PROFESOR

El proyecto debe ser **APROBADO** y recibir **máximo apoyo**.

Representa exactamente el tipo de aplicación innovadora de geoinformática que el curso busca promover. La combinación de:

- Procesamiento masivo de datos satelitales
- Machine learning espacial avanzado
- Problema de alto impacto económico
- Solución técnicamente sofisticada

Lo convierte en un proyecto ejemplar que podría:

1. Publicarse en conferencias de energía y geoinformática
2. Convertirse en startup o spin-off universitario
3. Servir como caso de estudio para futuras generaciones

Calificación sugerida: 7.0 (con potencial de nota máxima según ejecución)

7.3. Siguientes Pasos

1. **Inmediato:** Reunión con el estudiante para afinar alcance
2. **Semana 1:** Configurar accesos a GEE y datos del CEN
3. **Semana 2:** Definir arquitectura técnica detallada
4. **Semana 3:** Iniciar desarrollo con prototipo mínimo viable
5. **Mensual:** Revisiones de avance con feedback técnico

8. Referencias Técnicas Sugeridas

1. Gorelick, N. et al. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*.
2. Pérez-Arriaga, I. (2013). Regulation of the Power Sector. Springer.
3. Hong, T. et al. (2016). Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*.
4. Coordinador Eléctrico Nacional (2024). Informe de Operación del Sistema Eléctrico Nacional.
5. Zhang, Y. et al. (2023). Deep learning for renewable energy forecasting: A taxonomy, and systematic literature review. *Journal of Cleaner Production*.

6. Chilean Energy Ministry (2024). Política Energética Nacional 2050.

Dr. Francisco Parra O.
Profesor Curso Geoinformática
Universidad de Santiago de Chile
francisco.parra.o@usach.cl

Agosto 2025