

Análisis de Ideas de Proyecto

Evaluación de Componente Geoespacial

Curso: Geoinformática

Profesor: Francisco Parra O.

Agosto 2025

CRITERIO DE EVALUACIÓN

Este análisis evalúa cada proyecto según su potencial para incorporar análisis geoespacial significativo, considerando que es un curso de Geoinformática.

Alto: Componente espacial esencial (70-100 %)

Medio: Componente espacial relevante (30-70 %)

Bajo: Componente espacial marginal (¡30 %)

1 Resumen Ejecutivo

#	Proyecto	Potencial Geo	Recomendación
1	Postulación a empleos	Medio	Con ajustes
2	Monitoreo de stock	Medio	Con ajustes
3	Reclutamiento	Bajo	No recomendado
4	Comunicación interna	Bajo	No recomendado
5	Análisis publicitario	Medio	Con ajustes
6	Entregas a domicilio	Alto	Recomendado

2 Análisis Detallado

2.1 Proyecto 1: Postulación a Empleos según Destrezas

Potencial Geoespacial: **MEDIO** (40-50 %)

2.1.1. Componente Espacial Posible

- **Matching geográfico:** Conectar candidatos con empleos cercanos
- **Análisis de commute:** Tiempo de viaje, rutas óptimas, transporte público
- **Mapas de calor:** Demanda laboral por zona y especialidad
- **Movilidad laboral:** Patrones de desplazamiento trabajo-hogar
- **Clusters de industrias:** Identificar polos de desarrollo por sector

2.1.2. Propuesta de Mejora Geoespacial

“Sistema de Matching Laboral Geointeligente para Santiago”

```
# Ejemplo: Score de match considerando distancia y transporte
def calcular_score_match(candidato, empleo):
    # Factor distancia
    dist_km = calcular_distancia(candidato.ubicacion, empleo.ubicacion)
    score_dist = 1 / (1 + dist_km/10) # Decae con distancia

    # Factor transporte p blico
    tiempo_metro = calcular_tiempo_metro(candidato.ubicacion, empleo.ubicacion)
    score_transporte = 1 / (1 + tiempo_metro/30) # Decae con tiempo

    # Factor habilidades
    score_skills = calcular_match_habilidades(candidato.skills, empleo.requerimientos)

    # Score ponderado
    score_final = (score_skills * 0.5 +
                  score_dist * 0.3 +
                  score_transporte * 0.2)

    return score_final
```

2.1.3. Datos Geoespaciales Necesarios

- Red de transporte público (Metro, Transantiago)
- Ubicación de empresas por rubro
- Datos censales de población activa
- Tiempos de viaje reales (Google Maps API)

2.1.4. Limitaciones

- El matching de habilidades no es inherentemente espacial
- Muchos trabajos son remotos/híbridos post-pandemia
- Privacidad de datos de candidatos

2.2 Proyecto 2: Monitoreo de Stock

Potencial Geoespacial: **MEDIO (45-55 %)**

2.2.1. Componente Espacial Posible

- **Gestión multi-sucursal:** Optimización de inventario distribuido
- **Predicción espacial de demanda:** Por zona geográfica
- **Reabastecimiento inteligente:** Desde bodega más cercana
- **Análisis de competencia:** Stock según densidad de competidores
- **Logística de distribución:** Rutas óptimas de reposición

2.2.2. Propuesta de Mejora Geoespacial

“Sistema de Inventario Distribuido Geoptimizado”

```
# Ejemplo: Predicción de demanda por zona
import geopandas as gpd
from sklearn.ensemble import RandomForestRegressor

def predecir_demanda_espacial(producto, zona_gdf):
    features = []

    # Features espaciales
    features.append(zona_gdf['poblacion'])
    features.append(zona_gdf['ingreso-promedio'])
    features.append(zona_gdf['densidad-comercial'])
    features.append(calcular_distancia_competidor_cercano(zona_gdf))
    features.append(obtener_demanda_historica_vecinos(zona_gdf))

    # Features temporales
    features.append(dia_semana)
    features.append(es_quincena)
    features.append(temperatura_promedio)

    # Modelo predictivo
    modelo = RandomForestRegressor()
    demanda_predicha = modelo.predict(features)

    return demanda_predicha

def optimizar_reabastecimiento(sucursales_gdf, bodegas_gdf, demanda_predicha):
    # Problema de transporte con restricciones espaciales
    # Minimizar: costo-transporte + costo-stockout
    # Sujeto a: capacidad-bodegas, tiempos-entrega
    pass
```

2.2.3. Valor Agregado Geoespacial

- Reducción de quiebres de stock por zona
- Optimización de rutas de distribución
- Predicción de demanda localizada
- Balanceo de inventario entre sucursales

2.2.4. Limitaciones

- Requiere múltiples ubicaciones físicas
- El core del problema es más de optimización que espacial
- Necesita integración con sistemas ERP existentes

2.3 Proyecto 3: Automatización de Reclutamiento

Potencial Geoespacial: BAJO (10-20 %)

2.3.1. Componente Espacial Limitado

- Filtro por ubicación de candidatos (muy básico)
- Programación de entrevistas considerando zonas horarias
- Análisis de procedencia de candidatos

2.3.2. Por Qué No es Adecuado para Geoinformática

1. **Core no espacial:** El problema principal es procesamiento de texto y automatización
2. **Geografía marginal:** La ubicación es solo un filtro, no análisis
3. **Sin análisis espacial:** No requiere operaciones GIS complejas
4. **Mejor como proyecto de NLP/ML:** Parsing de CVs, matching de skills

2.3.3. Alternativa Sugerida

Si el grupo está interesado en RRHH, podrían pivotear a:

- “Análisis espacial de talento tech en Santiago”: Mapear dónde viven los desarrolladores, diseñadores, etc.
- “Optimización de oficinas satélite”: Dónde ubicar oficinas para minimizar commute

2.4 Proyecto 4: Comunicación Interna Empresarial

Potencial Geoespacial: BAJO (5-15 %)

2.4.1. Componente Espacial Muy Limitado

- Visualización de estructura organizacional por ubicación (si hay múltiples oficinas)
- Dashboard de métricas por región
- Coordinación entre oficinas en diferentes zonas horarias

2.4.2. Por Qué No es Adecuado

1. **Problema organizacional:** No geográfico
2. **Sin datos espaciales:** La comunicación es digital, no física
3. **Herramientas existentes:** Slack, Teams, etc. ya resuelven esto
4. **Fuera del scope:** Más apropiado para gestión de proyectos

2.4.3. Recomendación

NO recomendado para este curso. El grupo debería considerar otros problemas con componente espacial real.

2.5 Proyecto 5: Análisis de Métricas Publicitarias

Potencial Geoespacial: **MEDIO (35-45 %)**

2.5.1. Componente Espacial Posible

- **Geomarketing:** Análisis de efectividad por zona geográfica
- **Publicidad exterior:** Optimización de ubicación de vallas/pantallas
- **Segmentación geográfica:** Personalización de campañas por barrio
- **Attribution espacial:** Relacionar ventas físicas con ads digitales
- **Footfall analysis:** Medir tráfico en tiendas post-campaña

2.5.2. Propuesta de Mejora Geoespacial

“Sistema de Geomarketing y Attribution Espacial”

```
# Ejemplo: Attribution de campa a digital a ventas f sicas
def analizar_impacto_espacial_campana(campana_data, ventas_data, tiendas_gdf):
    # Crear zonas de influencia por tienda
    for tienda in tiendas_gdf.iterrows():
        zona_influencia = tienda.geometry.buffer(2000) # 2km radio

        # Usuarios impactados en la zona
        usuarios_zona = campana_data[
            campana_data.within(zona_influencia)
        ]

        # Correlacionar con ventas
        ventas_pre = ventas_data[
            (ventas_data.tienda_id == tienda.id) &
            (ventas_data.fecha < campana.fecha_inicio)
        ].mean()

        ventas_post = ventas_data[
            (ventas_data.tienda_id == tienda.id) &
            (ventas_data.fecha > campana.fecha_inicio)
        ].mean()

        lift = (ventas_post - ventas_pre) / ventas_pre

        # Modelo de attribution
        attribution_score = calcular_attribution(
            usuarios_impactados=len(usuarios_zona),
            lift_ventas=lift,
            distancia_promedio=usuarios_zona.distance(tienda.geometry).mean()
        )

    return attribution_scores
```

2.5.3. Datos Necesarios

- Ubicación de usuarios (anonimizada)
- Puntos de venta físicos
- Datos de campañas digitales con geotargeting
- Ventas por ubicación y tiempo

2.5.4. Limitaciones

- Privacidad de datos de usuarios
- Muchas métricas son puramente digitales (CTR, conversiones online)
- Requiere empresa con presencia física y digital

2.6 Proyecto 6: Optimización de Entregas a Domicilio

Potencial Geoespacial: **ALTO (80-90 %)**

2.6.1. Componente Espacial Core

- **Ruteo optimizado:** VRP (Vehicle Routing Problem) con restricciones
- **Predicción de tiempos:** Basado en tráfico real-time
- **Zonificación dinámica:** Asignación de repartidores por zona
- **Last-mile optimization:** Rutas peatonales/ciclistas
- **Hub location:** Ubicación óptima de centros de distribución
- **Tracking en tiempo real:** Seguimiento GPS de entregas

2.6.2. Propuesta Completa

“Sistema Inteligente de Logística Última Milla con IA Espacial”

```
import osmnx as ox
import networkx as nx
from ortools.constraint_solver import pywrapcp

class SistemaEntregasGeoptimizado:
    def __init__(self, area_servicio):
        self.graph = ox.graph_from_place(area_servicio, network_type='drive')
        self.delivery_points = []
        self.vehicles = []

    def optimizar_rutas_dia(self, pedidos_gdf, vehiculos_disponibles):
        # 1. Clustering espacial de pedidos
        clusters = self.clusterizar_pedidos(pedidos_gdf)

        # 2. Asignación vehículo-cluster
        asignaciones = self.asignar_vehiculos(clusters, vehiculos_disponibles)

        # 3. Optimización de ruta por vehículo
        rutas = {}
        for vehiculo, pedidos in asignaciones.items():
            ruta_optima = self.resolver_vrp(vehiculo, pedidos)
            rutas[vehiculo] = ruta_optima

        # 4. Balanceo dinámico
        rutas_balanceadas = self.balancear_cargas(rutas)

        return rutas_balanceadas

    def predecir_tiempo_entrega(self, origen, destino, hora_salida):
        # Considerar tráfico histórico y en tiempo real
        trafico_historico = self.get_trafico_historico(hora_salida)
        trafico_actual = self.get_trafico_realtime()

        # Ruta más rápida con pesos dinámicos
        ruta = nx.shortest_path(self.graph, origen, destino,
                               weight=lambda u,v,d: d['length'] * trafico_actual[u,v])

        tiempo_estimado = self.calcular_tiempo_ruta(ruta, trafico_actual)

        # Factor de incertidumbre
        buffer = self.calcular_buffer_tiempo(hora_salida, destino)

        return tiempo_estimado + buffer

    def monitorear_entregas_realtime(self):
        # Dashboard con mapa en vivo
        for vehiculo in self.vehicles:
            posicion_actual = vehiculo.get_gps_position()
            proxima_entrega = vehiculo.proxima_parada
            tiempo_restante = self.predecir_tiempo_entrega(
                posicion_actual, proxima_entrega, now()
            )

            # Alertar si hay retraso previsto
            if tiempo_restante > vehiculo.ventana_tiempo:
                self.notificar_cliente(proxima_entrega, nuevo_estimado)
                self.reasignar_si_necesario(vehiculo, proxima_entrega)
```


2.6.3. Tecnologías y Datos

- **Routing:** OSRM, GraphHopper, OR-Tools
- **Tráfico real-time:** Google Maps API, Waze API
- **Optimización:** Algoritmos genéticos, Simulated Annealing
- **Tracking:** GPS/GNSS, WebSockets para real-time
- **Predicción:** ML con features espaciotemporales

2.6.4. Métricas de Éxito

- Reducción tiempo promedio de entrega: 20-30 %
- Aumento entregas por ruta: 15-25 %
- Reducción km recorridos: 15-20 %
- Satisfacción cliente (entregas a tiempo): ¿95 %
- ROI: 6-12 meses

2.6.5. Por Qué es Ideal para Geoinformática

1. **100 % espacial:** Todo el problema es geográfico
2. **Datos reales disponibles:** OSM, APIs de tráfico
3. **Impacto medible:** Métricas claras de mejora
4. **Tecnologías GIS:** Routing, geocoding, spatial analysis
5. **ML espacial:** Predicción con features geográficas
6. **Visualización:** Mapas, dashboards, tracking

3 Recomendaciones por Potencial

3.1 Proyectos Recomendados

Proyecto 6: Sistema de Entregas

- Problema 100 % espacial
- Datos disponibles y accesibles
- Alto impacto comercial
- Múltiples técnicas GIS aplicables
- Excelente para portfolio

3.2 Proyectos con Potencial (Requieren Ajustes)

Proyecto 1: Matching Laboral

- Agregar análisis de movilidad urbana
- Incorporar factores de transporte público
- Mapeo de clusters industriales

Proyecto 2: Gestión de Inventario

- Enfocar en cadena de suministro multi-local
- Predicción de demanda geolocalizada
- Optimización de rutas de reabastecimiento

Proyecto 5: Geomarketing

- Centrarse en attribution offline-online
- Optimización de publicidad exterior
- Análisis de footfall y conversión

3.3 Proyectos No Recomendados

Proyecto 3: Reclutamiento

Proyecto 4: Comunicación Interna

- Componente espacial marginal o inexistente
- Mejor para otros cursos (IA, Sistemas de Información)
- No aprovechan las herramientas GIS del curso

4 Guía para Fortalecer Componente Espacial

4.1 Preguntas Clave

Para evaluar si un proyecto tiene suficiente componente espacial:

1. ¿El **dónde** es fundamental para resolver el problema?
2. ¿Se requieren operaciones espaciales (buffer, overlay, routing)?
3. ¿Los datos principales tienen coordenadas o direcciones?
4. ¿El análisis espacial agrega valor significativo a la solución?
5. ¿Se pueden aplicar técnicas de GIS/Remote Sensing?

4.2 Cómo Agregar Componente Espacial

Componente Débil	Fortalecimiento
Filtro por ubicación	Análisis de accesibilidad multimodal
Lista de direcciones	Geocoding + análisis de patrones espaciales
Distancia simple	Routing con restricciones + tráfico
Mapa estático	Dashboard interactivo con análisis
Datos puntuales	Series espaciotemporales
Zonas fijas	Zonificación dinámica optimizada

4.3 Stack Tecnológico Recomendado

Backend:

- Python: GeoPandas, Shapely, Folium
- PostGIS para base de datos espacial
- QGIS para análisis exploratorio
- OR-Tools para optimización

Frontend:

- Leaflet/Mapbox para mapas web
- Streamlit para dashboards rápidos
- D3.js para visualizaciones custom
- WebSockets para real-time

5 Conclusiones

Recomendación Final:

1. **Primera opción:** Proyecto 6 (Entregas) - Usar tal cual
2. **Segunda opción:** Proyecto 1 (Empleo) - Con enfoque en movilidad urbana
3. **Tercera opción:** Proyecto 2 (Inventario) - Como sistema multi-sucursal
4. **Evitar:** Proyectos 3 y 4 - No tienen componente espacial significativo

Recordar: Este es un curso de **Geoinformática**. El componente espacial debe ser central, no periférico. Si pueden resolver el problema sin mapas o análisis espacial, probablemente no es adecuado para este curso.