

Documentación Técnica : NoNA

ATRIBUCIÓN ACADÉMICA

Algoritmo de prefactibilidad económica desarrollado por Samuel Rodríguez Matus

Tesis de Maestría en Arquitecturas Avanzadas por la Universidad de Monterrey.

Asesorado por: **Arq. M.S.c Andrés Obregón López.**

Este proyecto calcula la viabilidad de proyectos habitacionales mediante análisis normativo, costos directos/indirectos, usos mixtos, estacionamiento y simulación de rentabilidad. Uso exclusivamente académico.

1. Descripción General

Este algoritmo, implementado en Python para el entorno Rhino/Grasshopper, realiza un análisis integral de viabilidad financiera y normativa para desarrollos inmobiliarios. El sistema procesa geometría arquitectónica y variables económicas para calcular áreas vendibles, costos de edificación, ingresos potenciales y métricas de retorno de inversión (ROI), incorporando módulos de simulación para optimización de precios según objetivos de utilidad.

2. Dependencias del Sistema

El script requiere las siguientes librerías para su ejecución en el entorno de Rhino Python:

- **Rhino.Geometry** (**rg**): Motor de cálculo geométrico para superficies y volúmenes.
- **scriptcontext** (**sc**): Interfaz de comunicación con el documento activo de Rhino.
- **math**: Librería estándar para operaciones matemáticas avanzadas.

- **csv, os:** Módulos para la gestión de directorios y exportación de reportes de datos estructurados.

3. Parámetros de Entrada (Inputs)

El algoritmo requiere la inyección de datos a través de los siguientes nodos en Grasshopper:

3.1 Geometría y Terreno

- **area_terreno** (Geometría: Surface/Brep): Superficie base del polígono del terreno.
- **valor_terreno** (Float): Valor económico base del suelo (\$/m²).
- **curva_retiros** (Geometría: Curve): Límite físico para el desplante del edificio (área útil tras retiros).

3.2 Normativa Urbanística

- **COS** (Float): Coeficiente de Ocupación del Suelo (0.0 - 1.0).
- **CAS** (Float): Coeficiente de Absorción del Suelo (0.0 - 1.0).
- **CUS** (Float): Coeficiente de Utilización del Suelo (Factor multiplicador).
- **Altura_Entrepisos** (Float): Altura libre promedio entre niveles (m).
- **H_Max** (Float): Restricción de altura máxima vertical (m).

3.3 Programa Arquitectónico

- **demolicion** (Bool) & **area_demolicion** (Float): Cálculo de preexistencias.
- **usos_mixtos** (Bool): Activación de zócalo comercial en planta baja.
- **num_locales** (Int): Subdivisión de unidades comerciales.
- **n_viviendas** (Int): Densidad habitacional proyectada (número de unidades).
- **areaCirculacionPorcentaje** (Float): Factor de eficiencia de planta (ej. 0.15 para 15% de circulaciones).

3.4 Costos y Factibilidad

- **estacionamiento** (Bool): Módulo de cálculo de cajones y sótanos por dotación normativa.

- `costoMetroConstruccion` (Float): Costo directo de edificación (\$/m²).
- `Costo_de_venta_m2` (Float): Valor de mercado proyectado para venta (\$/m²).

3.5 Simulación Financiera

- `utilidadDeseada` (Float): Margen de utilidad neta objetivo (%).
- `correrSimulacion` (Bool): Algoritmo iterativo para ajuste automático de precios.
- `Reporte_excel` (Bool): Exportación de resultados a CSV.

4. Lógica Algorítmica

4.1 Cálculo Normativo

El sistema determina los techos constructivos basándose en los coeficientes urbanos:

$$$$ COS_{\{m2\}} = Area_{\{terreno\}} * COS $$$$

$$$$ CUS_{\{m2\}} = Area_{\{terreno\}} * CUS $$$$

$$$$ Niveles = H_{\{max\}} / Altura_{\{entrepiso\}} $$$$

4.2 Estructura de Costos

Se integran costos directos (materiales y mano de obra) e indirectos porcentuales:

- **Costos Directos:** Edificación principal + Demoliciones + Urbanización.
- **Costos Indirectos:** Honorarios (15%), Legales (2%), Administrativos (10%), Financieros (3%), Comerciales (6%).

[!WARNING]

Nota de revisión de código: Se detectó una posible redundancia en la suma de la variable `cus_area` en la fórmula de costos directos original:

`costos_directos = cus_area * costoMetroConstruccion + total_dem + cus_area * costoMetroConstruccion`. Se sugiere revisión en la versión v1.0.

4.3 Módulo de Estacionamiento

Calcula la demanda de cajones basándose en la zonificación (Centro, Poniente, etc.) y la normativa vigente:

- **Demanda Vivienda:** $$$$N_{\{viviendas\}} * Factor_{\{distrito\}}$$$$

- **Demanda Comercio:** $\$Area_{\{Comercial\}} / Factor_{\{Uso\}}\$$

4.4 Algoritmo de Simulación (Goal Seeking)

Si la utilidad calculada es menor a la `utilidadDeseada`, el script ejecuta un bucle `while`:

1. Incrementa el *Precio Total de Venta* en pasos de \$1,000,000.
2. Recalcula la utilidad neta en cada iteración.
3. Se detiene al alcanzar el objetivo o el límite de iteraciones (150,000 ciclos), previniendo bucles infinitos.

5. Salida de Datos (Outputs)

El componente entrega variables atómicas para visualización en Grasshopper y genera un archivo externo `resultados_prefactibilidad.csv` con el resumen ejecutivo de la inversión.

Documentación generada para fines académicos - Universidad de Monterrey - 2026