

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



**“INTERPRETACIÓN DIGITAL DE PLANOS
ARQUITECTÓNICOS PARA REVISIÓN
AUTOMATIZADA RESPECTO A REGULARIZACIONES
DE CONSTRUCCIÓN CHILENA”**

FELIPE ALBERTO ENRIQUE NAREA GAGO

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA**

Profesor Guía: Mauricio Figueroa
Profesor Correferente: Eduardo Valenzuela

julio – 2024

DEDICATORIA

Para mi padre el cual siempre me ha hecho admirar la creatividad y la imaginación como motor de la actividad humana y como, muchas veces, las trabas y tiempos institucionales bloquean y ralentiza el trabajo. Para mi madre, quien en su eterna paciencia a formado la persona quien soy hoy y transmitido su curiosidad y labia. Para toda mi familia, quienes siempre estimularon mi creatividad e interés por la integridad del saber. También, para mi pareja y mis amigos quienes han estado presentes para limpiar mis lágrimas durante estos arduos meses. Finalmente, para mis perros, Rocco y Dominga, quienes me acompañaron fielmente por 12 años y quienes les agradezco haber compartido su pureza en mi vida hasta que sus cuerpos fallecieron durante el proceso de esta memoria, aunque *sus almas siempre me acompañarán.*

AGRADECIMIENTOS

Quiero agradecer a mi alma mater y todos los profesores, profesionales y administradores quienes realmente me han otorgado la oportunidad de llegar a desarrollar mi trabajo final de grado. Los años que ocupé en esta universidad han sido parte de los mejores años de mi vida, en ella he podido conocer a decenas de grandes profesionales y personas, he podido encontrar mi vocación y sobre todo me he encontrado a mí mismo. Entré como un niño y salí como uno de los profesionales que siempre he admirado y ello ha sido posible gracias a cada experiencia y relación que he tenido con cada persona con la que he tenido la suerte de cruzar. Ha sido gracias a cada derrota, a cada lagrima, a cada noche desvelada que he llegado más lejos de lo que creía posible durante los últimos años.

Realmente esta universidad bendice a aquellos quienes se adaptan a su agitada vida con su cultura particular, sus foros, con las actividades extracurriculares, y sobre todo con la dedicación de cientos de profesores quienes pocas veces reciben un salario adecuado a su esfuerzo.

RESUMEN

En el presente escrito, se analizará el cuello de botella del sistema público en la revisión de proyectos arquitectónicos necesarios, para otorgar permisos de edificación y recepción final, entre otros.

Con la finalidad de mejorar los procesos realizados por los revisores, se propondrá una solución de verificación de cumplimiento automatizada enfocada en la arquitectura 4.0, utilizando estándares de Modelado de Información de Construcción (BIM). El objetivo es desarrollar un conjunto de herramientas orientadas a crear programas capaces de fiscalizar el cumplimiento normativo mediante reglas codificadas y modelos 3D. Todo lo anterior busca mejorar la eficacia y eficiencia del proceso actual de revisión en las Direcciones de Obras Municipales (DOM), explorando el potencial de un motor orientado exclusivamente al sector de ingeniería, arquitectura y construcción. Se creará un motor que permita modelar algunos de los requisitos dados por el certificado de informes previos, con este se realizará una prueba de conceptos con un profesional con tal de validarlo y se explorará el potencial y usabilidad de dicho prototipo.

Keywords- *verificación de cumplimiento automatizada, arquitectura 4.0, normativas y regulaciones construcción chilenas, Modelado de información de construcción (BIM).*

ABSTRACT

At this document, the bottleneck of the public system in the review of architectural projects necessary to grant building permits and final receptions, among others, will be analyzed.

To improve the processes conducted by the reviewers, a solution for automated compliance verification focused on Architecture 4.0 will be proposed, using Building Information Modeling (BIM) standards. The goal is to develop a set of tools aimed at creating programs capable of enforcing regulatory compliance through codified rules and 3D models. All of this looks to improve the effectiveness and efficiency of the current process, exploring the potential of an engine exclusively oriented towards the engineering, architecture, and construction sector.

A motor will be created that allows modeling some of the requirements given by the certificate of prior reports, with which a proof of concept will be carried out with a professional in order to validate it and explore the potential and usability of said prototype.

Keywords— *Automated compliance verification, architecture 4.0, chilean buildings' rules and regulation, Building Information Modeling.*

GLOSARIO

- **DOM:** Dirección de Obras Municipales.
- **MINVU:** Ministerio de Vivienda y Urbanismo.
- **LGUC:** Ley General de Urbanismo y Construcción.
- **OGUC:** Ordenanza General de Urbanismo y Construcción.
- **ARC:** *Automated rule cheking.*
- **AIC:** Industria de arquitectura, ingeniería y construcción.
- **AISC:** *American institute of steel construction.*
- **SASE:** *Standard, analysis, synthesis and expression.*
- **IFC:** *Industrial foundation classes.*
- **BIM:** *Building Information Modelling.*
- **CIP:** Certificado de Informaciones Previos.
- **DSL:** *Domain Specific Language.*
- **Regex:** Expresiones regulares o *Regular expression*

INDICE DE CONTENIDOS

RESUMEN.....	iv
ABSTRACT	iv
INDICE DE CONTENIDOS	vi
INDICE DE FIGURAS.....	viii
INDICE DE TABLAS.....	ix
INTRODUCCIÓN	1
1.1 Definición del problema:	2
1.2 Estado del Arte:	5
1.2.1 Contexto del problema:.....	5
1.2.2 Digitalización DOM:	7
1.2.3 Metodología actual revisión de planos.	8
1.3 Objetivos de una solución:	10
1.3.1 Generales:.....	10
1.3.2 Específicos:	10
1.3.3 Alcance:	11
CAPÍTULO 2: MARCO CONCEPTUAL.....	12
2.1 Conceptos BIM:	12
2.2 IFC (Industry Foundation Classes):	13
2.3 Lenguaje de dominio específico (DSL):.....	14
2.4 Representación y procesamiento de objetos 3D:	16
CAPÍTULO 3: PROPUESTA DE SOLUCION	18
3.1 Introducción:	18
3.2 Ingeniería de Software:	19
3.2.1 Prácticas y Principios de Ingeniería de Software.....	20
3.3 Metodología de Desarrollo:.....	21
3.3.1 Enfoque Iterativo y Adaptativo:	21
3.3.2 Fases del Ciclo de Vida del Desarrollo:	21
3.3.3 Roles y Responsabilidades:.....	22
3.3.4 Técnicas y Herramientas Utilizadas:.....	22
3.3.5 Librerías y Tecnologías Específicas:	22
3.3.6 Pruebas y Validación:	23
3.4 Requisitos del Sistema:.....	23
3.4.1 Requisitos Funcionales	23
3.4.2 Requisitos No Funcionales.....	27
3.5 Arquitectura Técnica:	28
3.5.1 Visión General de la Arquitectura	29
3.5.2 Componentes del Sistema	29

3.6	Representación de Reglas:	31
3.7	Sintaxis de reglas:	31
3.8	Chequeo de propiedades:	36
3.9	Chequeo de relaciones:	36
3.10	Chequeo de composición lógica:	37
3.11	Evaluación de regla:	39
3.11.1	Identificación de objetos relevantes	39
3.11.2	Evaluación de la instancia matriz	41
3.12	Asignación, ejecución y construcción de objetos reales y virtuales:	42
3.12.1	Manejo de variables:	42
3.12.2	Ejecución de instrucciones	43
3.13	Editor de reglas:	45
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN.....		46
4.1	Validación de software:	48
4.1.1	Descripción técnica:	48
4.1.2	Declaraciones de normas:	49
4.2	Validación de hipótesis:	59
4.2.1	Interpretación digital de planos:	60
4.2.2	Contraste de proyecto con normas	63
4.2.3	Ejecución automática y salida del programa.....	64
4.3	Validación con prueba de conceptos:	66
4.3.1	Resultados cuantitativos:	67
4.3.2	Resultados cualitativos:	69
CAPÍTULO 5: CONCLUSIONES.....		71
5.1	Resumen de los Objetivos Alcanzados	71
5.2	Progresos de la “permisología”:	72
5.3	Trabajo Futuro:	74
5.4	Palabras finales del autor:	75
ANEXOS.....		76
Anexo A	Acta prueba de concepto	76
REFERENCIAS BIBLIOGRÁFICAS.....		89

INDICE DE FIGURAS

Figura 1: Flujo de solicitud DOM en línea.....	8
Figura 2: Diagrama arquitectura básica	31
Figura 3: Ejemplo distanciamientos y rasantes	32
Figura 4: Ejemplo árbol relacional	39
Figura 5: Regla Altura Máxima	50
Figura 6: Ejemplo Altura máxima Incumple	50
Figura 7: Proyecto Revit Max altura no cumple	51
Figura 8 Proyecto Revit Max altura Cumple.....	51
Figura 9: Regla Antejardín	52
Figura 10: Ejemplo antejardín incumple	53
Figura 11: Proyecto Antejardín No cumple	53
Figura 12: Proyecto Antejardín Cumple	54
Figura 13: Regla de Rasantes.....	55
Figura 14: Ejemplo incumplimiento de rasantes	55
Figura 15: Proyecto rasantes no cumplen	56
Figura 16: Proyecto que incumple rasantes con adosamiento	56
Figura 17: Ejemplo incumplimiento de Rasante con adosamiento.....	57
Figura 18: Regla Distanciamiento	58
Figura 19: Ejemplo incumplimiento distanciamiento	58
Figura 20: Proyecto que incumple Distanciamiento	59
Figura 21: Proyecto Reñaca obtenido desde Revit 2025	60
Figura 22: Proyecto digitalizado IFC	61
Figura 23: Clases IFC obtenidas	62
Figura 24: Ejemplo Visualizador elementos que incumplen normas de distanciamiento ...	64
Figura 25: Ejemplo Visualizador elementos que incumplen normas con referencia de objetos que sí cumplen.	65
Figura 26: Ejemplo fila de matriz evaluación cuya interacción entre los objetos resultan en incumplimiento de la norma	65
Figura 27: Evaluación de antejardín	68
Figura 28: Evaluación Rasantes	69

INDICE DE TABLAS

Tabla 1: Plazo de respuesta de las DOM	2
Tabla 2: Resumen practicas Ing. Software	21
Tabla 3: Requisito Funcional-01	23
Tabla 4: Requisito Funcional-02	24
Tabla 5: Requisito Funcional-03	24
Tabla 6: Requisito Funcional-04	25
Tabla 7: Requisito Funcional-05	25
Tabla 8: Requisito Funcional-06	26
Tabla 9: Requisito Funcional-07	26
Tabla 10: Requisito No Funcional-01.....	27
Tabla 11: Requisito No Funcional-02.....	27
Tabla 12: Requisito No Funcional-03.....	28
Tabla 13: Requisito No Funcional-04.....	28
Tabla 14: Distanciamientos hacia deslindes.....	32
Tabla 15: Primera sección regla Distanciamiento	33
Tabla 16: Segunda sección regla Distanciamiento	33
Tabla 17: Tercera sección regla Distanciamiento.....	33
Tabla 18: Cuarta sección regla Distanciamiento	34
Tabla 19: Quinta sección regla Distanciamiento	34
Tabla 20: Sexta sección regla Distanciamiento	34
Tabla 21: Lista de propiedades geométricas	36
Tabla 22: Lista de relaciones geométricas	37
Tabla 23: Ejemplo matriz combinatoria objetos	40
Tabla 24: Funciones de manipulación de objetos	43
Tabla 25: Descripciones Reglas a evaluar	47
Tabla 26: Clases IFC por validar	61
Tabla 27: Resultado prueba conceptos	67

INTRODUCCIÓN

De acuerdo con la legislación vigente, para comenzar y terminar la construcción de cualquier edificación se debe cumplir una serie de requisitos establecidos por la Ley General de Urbanismo y Construcción, y su respectiva Ordenanza (LGUC y OGUC), entre otras normativas. Estos requisitos son revisados, aprobados o rechazados por la Dirección de Obras Municipales (DOM), aunque conforme a la complejidad del proyecto pueden intervenir otros organismos. Es de conocimiento general que los DOM tienden a retrasarse en los plazos de revisión, situación que puede causar pérdidas económicas debido al costo oportunidad de los plazos en la construcción. Además, en dichas revisiones los criterios quedan al árbitro de un funcionario municipal, muchas veces cuyos criterios no siempre son acordes a la correcta interpretación de la legislación. Por lo anterior, es que se requiere nuevos métodos que agilicen los procesos de los DOM, de tal modo que se pueda reducir la carga de trabajo, manteniendo la eficacia dada la relevancia de los requerimientos de construcción en el territorio nacional. Por ello, se propone un enfoque basado en la arquitectura 4.0 para abordar esta cuestión, utilizando tecnologías de diseños que permitan la automatización parcial de la revisión de planos, con el objetivo de brindar un servicio preliminar de revisión que permita a los arquitectos reconocer los errores de sus proyectos antes de ser entregados a los DOM, y que estas últimas cuenten con un filtro sencillo para rechazar proyectos que no cumplan las regulaciones.

Este documento de memoria se estructura en varias secciones que detallan el proceso completo de desarrollo y validación del motor de revisión automática. En primer lugar, se presenta una revisión de la literatura y el estado del arte en el ámbito de la automatización en la revisión de proyectos arquitectónicos. A continuación, se describen los aspectos técnicos y metodológicos del desarrollo del software, incluyendo la implementación de tecnologías BIM y la codificación de reglas normativas.

La sección de validación expone los resultados de las pruebas de concepto realizadas, proporcionando una evaluación exhaustiva de la eficacia del motor de revisión. Finalmente, se presenta una discusión sobre los beneficios potenciales de esta tecnología para la burocracia en Chile, destacando cómo la automatización puede contribuir a la eficiencia, transparencia y uniformidad en la revisión de proyectos arquitectónicos.

A grandes rasgos, este estudio se propone una solución innovadora para los desafíos actuales de la obtención de permisos, además que, también sienta las bases para futuras investigaciones y desarrollos en la automatización de procesos normativos en la construcción. La integración de tecnologías avanzadas como BIM y la arquitectura 4.0 promete transformar significativamente el paisaje de la construcción en Chile, mejorando la calidad y rapidez en la aprobación de proyectos.

DEFINICIÓN DEL PROBLEMA

1.1 DEFINICIÓN DEL PROBLEMA:

En la legislación vigente, para iniciar y finalizar una edificación sin importar su envergadura, se requiere el permiso de edificación y recepción final respectivamente, cuyos requisitos se encuentran establecidos en la Ley General de Urbanismo y Construcciones [LGUC, 1976], además de la Ordenanza General de Urbanismo y Construcciones [OGUC, 1992], Ordenanzas de los Planos Reguladores Metropolitanos y Locales [Ministerio de Vivienda y Urbanismo, 1975]. La función de verificar y realizar el cumplimiento de dichas normas lo realiza la Dirección de Obras Municipales (DOM), aunque de acuerdo con la complejidad del proyecto, también pueden intervenir otras instituciones públicas, tales como: Dirección de Obras Hidráulicas, Dirección de Aeronáutica Civil, Ministerio de Medio Ambiente, Ministerio de Transportes (Dirección de Vialidad), Servicio Agrícola y Ganadero, Ministerio de Agricultura, Consejo de Monumentos Nacionales, Servicios de Salud, etc. Sin perjuicio de las consultas realizadas a la Contraloría General de la República o las distintas Secretarías Regionales Ministeriales de Vivienda.

La Ordenanza General de Urbanismos y Construcciones indica que el plazo de respuesta que las DOM deben cumplir es de 15 días y 30 días hábiles dependiendo de su clasificación

Clasificación	Plazo [días]
Anteproyectos en general	15
Proyectos obras menores	15
Proyectos en general	30
Proyectos en general que cuenten con el patrocinio de un revisor Independiente ¹	15

Tabla 1: Plazo de respuesta de las DOM
Fuente: Artículo 1.4.10 OGUC [MINVU, 1992]

El Revisor independiente es una figura creada con el propósito de evitar la congestión de proyectos al interior de las DOM, entregándole a este profesional (Arquitecto) la capacidad de revisar – y aprobar– las normas de la OGUC referentes a todas aquellas cuestiones de funcionamiento interior del proyecto (Por ej. Vías de escape, Accesibilidad Universal, ventilación, Resistencia al fuego, Acondicionamiento térmico, etc.) dejando solo los aspectos urbanísticos del proyecto para que sean revisados exclusivamente por las DOM (zonificación, constructibilidad, ocupación de suelo, alturas, rasantes, entre otras). Aún así, las DOM no consideran esta certificación y siguen revisando a pesar de lo expresado en la OGUC. [MINVU, 1992]

La creciente complejidad normativa para la aprobación de proyectos ha causado que la aprobación de estos pueda demorar de 10 hasta 12 meses en aquellos proyectos más complejos [Osorio, 2021], tales como hospitales, grandes complejos comerciales y habitacionales, etc. y en aquellos proyectos simples, tales como ampliaciones, viviendas unifamiliares, pequeños comercios, etc. Se generan atrasos de en promedio 2 a 4 meses, lo cual causa pérdidas económicas y de costo oportunidad a los solicitantes [Mercurio, 2022], sin mencionar la irritación de los solicitantes por lo que se podría considerar como un mal servicio.

También hay que mencionar la disparidad y dispersión de criterios en la aplicación de la normativa que se encuentra en las observaciones que hacen los arquitectos de las DOM [Herman, 2021], las que muchas veces son injustificadas y causan entorpecimiento en la aprobación de los expedientes de construcción, extendiendo innecesariamente el tiempo de aprobación más allá de lo establecido por la OGUC.

La participación de múltiples actores no coordinados, más la falta de capacitación y escasa cantidad de personal municipal [Minvu, 2021] para responder a la demanda por permisos ha causado este creciente y acumulativo problema, que afecta específicamente el sector construcción, que se sabe es un importante motor de la economía nacional.

Existen, a lo menos, dos niveles de proyectos afectados de acuerdo con el grado de complejidad de estos

1. **Proyectos Menores:** Proyectos de ampliaciones, regularizaciones, viviendas unifamiliares, edificios menores, comercio de menor escala (que no requieran aprobación de proyectos viales, ni la intervención de otras reparticiones públicas).

Estos proyectos, aunque representan pocos metros cuadrados construidos, son los que presentan mayor frecuencia de ingresos a las DOM, y su tiempo de tramitación se ve afectado por los proyectos de mayor envergadura, los que consumen principalmente recursos profesionales disponibles en las DOM.

2. **Proyectos complejos:** Necesariamente requieren de una revisión mayor, así como coordinación con otros permisos sectoriales. Grandes complejos de vivienda, edificios de gran altura, centros comerciales medios y grandes, aeropuertos, hospitales, etc.

Estos proyectos son los que representan la mayor superficie edificada, y son los que reflejan una tramitación de permisos que va hasta los 10 - 12 meses. Los proyectos complejos y de mayor envergadura han de contar, al ingresarlos a la DOM con la revisión de un Arquitecto Revisor Independiente, el cual certifica que el proyecto cumple con la normativa establecida en la LGUC y OGUC, pero excluyendo los aspectos urbanísticos, los cuales son de exclusiva incumbencia de la DOM.

Estas normas urbanísticas, en líneas generales, están informados en el Certificados de Informaciones Previas (CIP) que las DOM entrega al interesado al comenzar el diseño del proyecto, donde se conjugan tanto normas generales, como asimismo otras de carácter local (Planos Reguladores Comunales o Metropolitanos)

Es así como podemos decir que desocupando a los arquitectos municipales que hoy se encuentran destinados a proyectos menores, se podrá disminuir la carga laboral y por tanto los tiempos de respuesta municipal en los proyectos complejos, los cuales requieren un mayor control humano dada la jurisprudencia de las regulaciones para este tipo de proyectos.

Los retrasos en la aprobación de proyectos de construcción por parte de las Direcciones de Obras Municipales (DOM) han generado un impacto significativo en diversos ámbitos de la economía y la sociedad chilena. Estos atrasos, desencadenan una serie de consecuencias que afectan no solo a los solicitantes directos, sino también al sector de la construcción en su conjunto y, por extensión, a la economía nacional.

La industria de la construcción, reconocida como un motor fundamental de la economía chilena y que representa aproximadamente el 7% del PIB del país [Leon, 2023], se ve particularmente afectada por estos retrasos. La ralentización en la aprobación de proyectos genera un efecto dominó que impacta en la actividad del sector, afectando los niveles de empleo y el crecimiento económico general.

Asimismo, estos atrasos conllevan pérdidas económicas directas para los desarrolladores, quienes deben enfrentar costos adicionales como pagos de intereses extendidos, gastos de mantenimiento de propiedades sin uso y costos de oportunidad por ingresos no percibidos. Estos costos extras, inevitablemente, tienden a trasladarse al consumidor final, contribuyendo al aumento en los precios de las viviendas y propiedades comerciales, lo que a su vez puede dificultar el acceso a la vivienda para muchos chilenos.

La frustración generada por estos retrasos, que superan ampliamente los plazos establecidos por la OGUC de 15 a 30 días, no solo afecta a los solicitantes directos, sino que también erosiona la confianza ciudadana en las instituciones públicas. Esta pérdida de confianza puede desalentar futuras inversiones y proyectos de desarrollo, creando un círculo vicioso que afecta negativamente al sector y a la economía en general.

La sobrecarga en el sistema municipal, producto de la acumulación de proyectos pendientes de aprobación, genera una presión adicional sobre el ya escaso personal municipal. Esto no solo afecta a los proyectos privados, sino que también impacta en la aprobación de infraestructuras públicas cruciales como hospitales y escuelas, lo que puede tener consecuencias directas en la prestación de servicios públicos esenciales y en la calidad de vida de los ciudadanos.

Además, los extensos tiempos de aprobación pueden mermar la competitividad de Chile en el ámbito de la inversión inmobiliaria y de construcción a nivel regional. Según el informe *Doing Business* del Banco Mundial, Chile se ubica en el puesto 41 en el índice de facilidad para obtener permisos de construcción, por detrás de varios países latinoamericanos, teniendo un promedio de 12 procedimientos por proyecto y una demora de 195 días [worldbankgroup, 2024], lo que podría disuadir a potenciales inversores internacionales.

Esta cascada de consecuencias pone de manifiesto la urgencia de abordar los problemas en el proceso de aprobación de proyectos de construcción. La solución a esta problemática no solo mejoraría la eficiencia administrativa, sino que también tendría un impacto positivo en el desarrollo económico y social del país, beneficiando tanto a los actores directos del sector de la construcción como a la ciudadanía en general.

1.2 ESTADO DEL ARTE:

1.2.1 CONTEXTO DEL PROBLEMA:

El problema del control automatizado de normas o conocido en inglés como “*automated rule checking* (ARC)” es un método basado en normas que provee de un control de los elementos de una construcción respecto a las regularizaciones del sector de construcción, que considera el tamaño, distribución, relaciones y características de los elementos propios de una edificación con sus respectivos códigos y normas. Este ha sido ampliamente estudiado en las últimas décadas con distintos grados de éxito, sin embargo, no se ha llegado a un consenso de cómo abarcar universalmente este problema dada la complejidad de la interpretación de reglas, la localidad de ellas y la resistencia de instituciones y empresas afiliadas a la industria de arquitectura, ingeniería y construcción (AIC) [Sobhkhiz, 2021].

El primer acercamiento a este problema ocurrió en la década de los 1960s, como un subcampo del estudio de la inteligencia artificial y lingüística para la investigación de la automatización de la comprensión y creación del lenguaje humano. Las especificaciones y normas del instituto americano de construcción en acero (AISC, por sus siglas en inglés) han sido creadas usando tablas de decisiones desde la década mencionada anteriormente. La investigación realizada por Fenves (1966), usando tablas de decisión es considerado el primer estudio en esta área de la literatura [Fenves, 1966]. En 1973, el AISC emprendió una reestructuración de sus especificaciones que derivó en la presentación de un modelo teórico en el que el conocimiento de una regulación podía ser explicado con una estructura de cuatro capas, modelado SASE (*Estandar, analysis, synthesis and expression*). Este modelo fue utilizado para desarrollar un software de procesamiento de conocimiento experto que se desarrolló por Fenves y su equipo en 1987, con el fin de proporcionar una estructura global para familias de códigos y normas relacionados en las numerosas jurisdicciones estadounidenses [Garret, 1987]. Delis y Delis [Delis, 1995] desarrollaron un sistema experto basado en conocimiento para la revisión automática del reglamento de incendios, donde su método propuesto incluía un conjunto de reglamentaciones en formato IF-THEN.

En el último tiempo ha existido un rápido progreso en los sistemas automatizados de comprobación en conformidad a los códigos de construcción internacionales [Lee, 2020]. Sin embargo, la inmensa mayoría de estos sistemas requieren la conversión manual de las normativas y códigos a formatos accesibles por ordenador, tales como tablas de decisión [Tan, 2010] modelos semánticos de los códigos de construcción [Macit, 2017] modelos de conocimiento [Dimyady, 2016] y reglas lógicas [Zhong, 2012]. Pauwles, propusieron un sistema automatizado de comprobación de la conformidad de los códigos basado en las “*industrial foundation classes*” (IFC) [Pauwels, 2011], un modelo de datos el cual define las características de los datos relacionados con el diseño, construcción, mantenimiento y operaciones de obras civiles, promovido como el estándar ISO 16739. Hjelseth y Nisbet [Hjelseth, 2010] propusieron un método de marcado manual para convertir las regulaciones de construcción de ser un texto normativo a declaraciones lógicas estructuradas. Beach [Beach, 2015] desarrollaron un sistema de comprobación de cumplimientos de las regulaciones basados en reglas que utilizan ontologías para mapear entidades en modelos IFC a conceptos de códigos de construcción. “*Solibri model checker*” (SMC) contiene como un complemento la capacidad de comprobación de la conformidad de los códigos que verifica modelos IFC, según un conjunto de normas generadas manualmente. El proyecto “*Construction and real state network*” (CORENET) es un sistema de comprobación de cumplimiento basado en IFC y respaldado por el gobierno de Singapur, siendo este, el encargado de proporcionar los códigos de construcción para comprobar los diseños de los edificios en un formato procesable por computadores.

Es importante para este escrito, mostrar a los lectores que avances han permitido que este tipo de trabajos se puedan realizar en su mayoría de casos. Para ello se requiere conversar sobre la modelización de la información de las construcciones o también denominado “BIM” (*Building Information Modelling*) y la consiguiente transformación digital en la industria de la construcción. BIM no solo promueve la creación de modelos 3D ricos en datos para todas las fases de un proyecto de construcción, desde el diseño hasta la gestión de las instalaciones, sino que también facilita la colaboración y la interoperabilidad entre las diferentes partes interesadas del proyecto. BIM ha hecho posible la generación de reglas automatizadas para la verificación de la conformidad con los códigos de construcción, al proporcionar un formato estructurado y estandarizado para los datos de construcción.

El desafío de interpretar y aplicar correctamente los códigos de construcción se reconoce ampliamente en la literatura. Según Eastman [Eastman, 2009], este problema se debe en gran parte a la naturaleza inherentemente compleja de los códigos de construcción y a la necesidad de un conocimiento especializado en arquitectura y diseño para su interpretación adecuada.

En cuanto a la digitalización de las normativas y códigos de construcción, Sobhkhiz resume los desafíos de ARC, donde considera que: (1) la información no pública o resumida necesaria para analizar una norma de un edificio, (2) la falta de transparencia en los procesos de un sistema ARC necesaria para el seguimiento iterativo de resolver el cumplimiento de directrices, (3) la falencia de una semántica estandarizada y computable en los reglamentos y (4) la resistencia al cambio de organizaciones públicas y privadas para adoptar tecnologías ARC [Sobhkhiz, 2021].

Finalmente, sobre el futuro prometedor de la verificación de conformidad con los códigos de construcción, Zhang [Zhang, 2014] entrega la sugerencia de que la inteligencia artificial y el aprendizaje automático podrían desempeñar un papel crucial en la mejora de este proceso, argumenta que estas tecnologías podrían permitir una interpretación de código más precisa y eficiente, así como la creación de sistemas de verificación más sofisticados y precisos.

1.2.2 DIGITALIZACIÓN DOM:

En Chile se ha llevado a cabo un proceso de automatización, modernización y digitalización de las DOM desde principios del milenio, aunque este ha sido desigual en términos territoriales dada la naturaleza municipal e independiente de las direcciones de obras, donde los municipios con mayor poder adquisitivo o crecimiento urbano han liderado los esfuerzos del país.

Aunque en la última década se ha impulsado un esfuerzo nacional para la digitalización de los DOM con la plataforma "DOM en línea" que permite efectuar los trámites de las direcciones de obras a través de internet. Liliane Etcheberrigaray, coordinadora Nacional de DOM en línea, comenta que, a la fecha de 2023, 208 de las 346 comunas nacionales ya contaban activamente con el sistema, con más de 80 trámites disponibles. Sin embargo, a partir de los reportes obtenidos desde la plataforma, se observó que desde 2019 hasta 2023 no había aumentado significativamente los niveles de uso del sistema. Además, la inmensa cantidad de trámites digitalizados se concentró en un 20% de las municipalidades [Construye2025, 2023].

Lo anterior, en parte, se explica dada la desigualdad que existe entre municipios en las dotaciones de personal, además de los recursos tecnológicos con los que cuentan. Afirma Yuri Rodríguez, Secretario General de la Asociación de Directores de Obras de Chile, que "Actualmente, hay una enorme diferencia entre las más de 300 comunas del país, en muchas de las cuales no se cuenta con las tecnologías y equipos para una buena gestión" [Construye2025, 2023].

Por otro lado, es importante destacar que la adopción de sistemas digitales y plataformas en línea no solo depende de los recursos económicos, sino también de la capacitación del personal municipal y la existencia de una cultura organizacional orientada a la innovación tecnológica.

1.2.3 METODOLOGÍA ACTUAL REVISIÓN DE PLANOS.

La metodología predominante utilizada por las Direcciones de Obras Municipales para la revisión de planos y expedientes de construcción se basa principalmente en la observación exhaustiva y el análisis manual por parte de profesionales capacitados. Sumado a ello, el actual flujo de trabajo para las solicitudes ingresadas en “DOM en línea” está representado en **Figura 1**, en el cual se puede distinguir 3 etapas con tal de obtener la aprobación o rechazo de una solicitud.

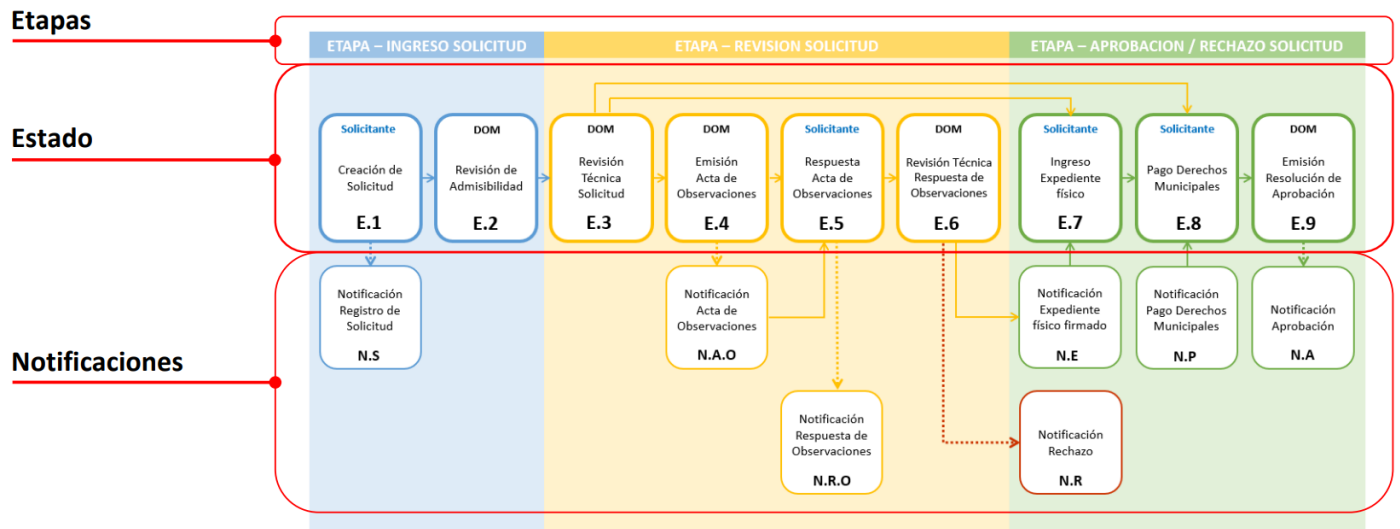


Figura 1: Flujo de solicitud DOM en línea
Fuente: Manual del solicitante (MINVU, 2021)

Siendo la primera etapa el “ingreso de la solicitud”, donde luego de la creación de una solicitud, el DOM en cuestión realiza una revisión de admisibilidad, el cual consiste en cerciorar que todos los documentos solicitados en una revisión se encuentren en el expediente, valga la redundancia, de la solicitud.

Hecho lo anterior, se pasa a la segunda etapa de “revisión solicitud”, la cual inicia cuando se declara admisible la solicitud y sus expedientes, siendo asignados a un arquitecto revisor, quien revisará en detalles los antecedentes presentados y el cumplimiento de la normativa aplicable, según el tipo de solicitud realizada. Cuando la solicitud corresponde a autorizaciones, aprobaciones, permisos, regularizaciones, etc. El arquitecto revisor elaborará un “acta de observaciones” cuando corresponda, el cual contiene todas las observaciones a corregir del proyecto analizado. Si el expediente no presenta observaciones, se elaborará, visará y firmará una “resolución de aprobación”. Excepcionalmente, las solicitudes acogidas a la Ley 20.898 (que establece un procedimiento simplificado para la regularización de viviendas de autoconstrucción), que no cumplan con los requerimientos para ser autorizada en primera instancia, se elaborará inmediatamente un “oficio de rechazo” sin existir “acta de observaciones”.

En la situación de existir un “acta de observaciones” el solicitante deberá resolver y enviar las correcciones de vuelta al DOM para continuar el proceso de revisión, una vez recibida el expediente corregido del solicitante, el arquitecto revisor recibirá el expediente y deberá verificar que las observaciones del acta hayan sido subsanadas.

En caso de cumplimiento, el arquitecto revisor redacta un borrador de la “resolución de aprobación”. Caso contrario, se redactará un borrador de “oficio de rechazo”, donde ambos documentos deberán ser visado por el jefe de unidad y firmado por el Directo de Obras Municipales.

Realizada la revisión técnica de la respuesta de las observaciones, da inicio la tercera y última etapa de “aprobación o rechazo solicitud” donde en caso de incumplir los requerimientos de aprobación y existir el “oficio de rechazo”, el expediente será rechazado finalizando el trámite.

En caso de cumplir los requisitos y existir la “resolución de aprobación” el solicitante deberá enviar los expedientes físicos, cuando se haya confirmado la recepción por parte del arquitecto revisor, se deberá realizar el pago de los derechos municipales correspondientes, el cual será validado por el revisor administrativo de la DOM. Finalmente, validado el pago, el Directo de Obras deberá firmar y emitir los documentos correspondientes a la aprobación de la solicitud finalizando el proceso.

Agregar a todo lo anterior que en caso de que el expediente posea la firma de un revisor independiente, el proceso de la segunda etapa se ve reducida a solo tener que revisar cuestiones urbanísticas, dando por aprobados automáticamente todos los otros requisitos de la revisión.

Desarrollando el proceso de revisión por parte del arquitecto revisor, consiste en una revisión minuciosa de todos los antecedentes técnicos presentados, incluyendo planos de arquitectura, estructuras, instalaciones, memorias de cálculo, especificaciones técnicas y demás documentación exigida por la normativa vigente. Los profesionales revisores, que suelen ser arquitectos, ingenieros y constructores civiles, deben verificar que el proyecto cumpla con la Ordenanza General de Urbanismo y Construcciones (OGUC), el Plan Regulador Comunal, las normas técnicas aplicables y cualquier otra reglamentación pertinente.

La revisión se lleva a cabo de forma manual, analizando cada uno de los planos y documentos entregados. Se realizan mediciones, cálculos y comprobaciones para detectar posibles errores, omisiones o incumplimientos normativos. Este proceso requiere un alto nivel de conocimientos y experiencia por parte de los revisores asignados.

Además de la revisión documental, en ciertos casos se pueden realizar visitas a terreno para constatar que las obras en ejecución se ajusten a los planos aprobados y cumplan con las condiciones del permiso de edificación.

Es menester referir que es en la segunda etapa en donde se produce el cuello de botella explicado en la definición del problema, ya que es el proceso de revisión manual hecho tanto por revisores municipales como independientes el que requiere el mayor uso de trabajo y tiempo, ya que dicho proceso incluye, aparte de la primera revisión, el tiempo utilizado para enviar correcciones, corregir por parte del arquitecto solicitante y verificar nuevamente que hayan sido subsanada, sin mencionar que puede existir un proceso aparte de apelación a las observaciones el cual ocupa aún más tiempo al tener que recurrir a otras agencias gubernamentales superiores al DOM con tal de aceptar o denegar la apelación y poder continuar el proceso de aprobación o rechazo de la solicitud.

Si bien actualmente la observación manual es la metodología predominante, existen iniciativas nacionales orientadas a impulsar la adopción de la metodología BIM en las DOM.

El programa "planBIM", impulsado por CORFO a 10 años plazo, tiene como una de sus metas la utilización de BIM para el desarrollo y operación de proyectos de edificación e infraestructura pública al año 2020, formando parte del programa "Construye 2025" [Construye2025, 2023]. Aunque aún no se ha implementado ampliamente, estas iniciativas buscan facilitar y modernizar los procesos de revisión de planos y proyectos en las municipalidades en el futuro.

1.3 OBJETIVOS DE UNA SOLUCIÓN:

1.3.1 GENERALES:

Crear un programa capaz de identificar los distintos objetos de un proyecto arquitectónico y que permita contrastarlos en cumplimiento de las normativas descritas por el certificado de informes previos.

1.3.2 ESPECÍFICOS:

1. Desarrollar e implementar una plataforma que permita ingresar planos digitalmente e interpretarlos.
2. Contrastar planos respecto a las exigencias descritas en el certificado de informes previos semejante a como si lo resolviese un revisor municipal o independiente.
3. Ejecutar evaluaciones automáticamente, entregando como respuesta un listado de observaciones del plano en un lenguaje que aporte al arquitecto proyectista la posibilidad de corregir su propuesta con conocimiento de causa.

1.3.3 ALCANCE:

Se asumirá que los planos entregados al sistema serán en formato IFC 4 Reference View y tendrán que cumplir con buenas prácticas del diseño de planos de casas en plataformas BIM según lo descrito en el Estándar BIM Para Proyectos Públicos del plan BIM [Planbim, 2023], pues el programa solo se encargará de constatar cumplimientos de normas técnicas y no normas de formas. Además, dado que para el desarrollo de esta memoria será necesario un prototipo de revisión automática, este se restringirá a recibir proyectos habitacionales menores con fin de demostrar la viabilidad de este. Finalmente se realizarán pruebas respecto a las normas: Distanciamientos y rasantes O.G.U.C 2.6.2 y 2.6.3 respectivamente, Altura máxima de edificación O.G.U.C 2.1.23 y Distanciamientos línea oficial y edificación según plan regulador descrito en el decreto 2732 (26-julio-2019) de la Municipalidad de Quilpué.

CAPÍTULO 2: MARCO CONCEPTUAL

Para poder comprender este trabajo, será estrictamente necesario entender con qué sistema serán interpretados los dibujos de diseños de planos arquitectónicos, cómo interactúan los objetos de un plano entre sí, y las propiedades que estos poseen. Por otra parte, se explicará cómo dicho sistema será traducido computacionalmente. También se explicará al lector cómo las normativas serán representadas en un lenguaje de dominio específico, que permita tanto el descifrado de ellas para el sistema de interpretación como facilitar la creación o modificación de reglas sin necesidad de contar con conocimientos técnicos computacionales complejos. Finalmente, se describirá la metodología necesaria para poder transcribir las reglas y normas del certificado de información previa al lenguaje de dominio específico.

2.1 CONCEPTOS BIM:

Según la asociación y autoridad buildingSMART (principal autoridad en la creación y adopción de estándares internacionales BIM), BIM es una metodología de trabajo colaborativa para la creación y gestión de un proyecto de construcción. Su objetivo es centralizar toda la información del proyecto en un modelo de información digital creado por todos sus agentes [buildingSMART, s.f.-a]. BIM supone la evolución de los sistemas de diseño tradicionales basados en el plano, ya que incorpora información geométrica (3D), de tiempos (4D), de costes (5D), ambiental (6D) y de mantenimiento (7D) [buildingSMART, s.f.-b]. El uso de BIM va más allá de las fases de diseño, abarcando la ejecución del proyecto y extendiéndose a lo largo del ciclo de vida del edificio, permitiendo la gestión de este y reduciendo los costes de operación [buildingSMART, s.f.-c]. Este modelo de información digital, o modelo BIM, es una representación rica en datos de las características físicas y funcionales de la edificación.

BIM permite generar modelos virtuales integrando información geométrica (3D), datos detallados de los componentes, y una amplia gama de información complementaria. Mediante BIM es factible almacenar, clasificar y relacionar todos los componentes que conforman una construcción, facilitando una gestión más eficiente y colaborativa del proyecto. Un modelo BIM se compone principalmente de tres elementos clave: objetos, propiedades y relaciones entre objetos [Feito, 2024].

- **Objetos:** Cada objeto representa un componente físico de la construcción, como por ejemplo muros, columnas, ventanas, etc. Posee un identificador único, una clasificación por tipo (clase), y una representación geométrica precisa. Por ejemplo, un objeto de tipo "Muro" con identificador "M-01" podría tener una geometría de 2x2x2.4 metros.

- **Propiedades:** Además de su geometría, los objetos BIM contienen un conjunto de propiedades o atributos que describen sus características específicas, como materiales, resistencia, costos, fabricante, etc. Estas propiedades van más allá de la mera representación visual, y son fundamentales para el análisis y cumplimiento de normas y regulaciones. Por ejemplo, el objeto "Muro M-01" podría tener una propiedad "Material: Hormigón armado". Por otro lado, se definen por los niveles de detalle del modelo (LOD), relacionados con el contenido gráfico de los modelos y los niveles de información del modelo (LOI), referenciados al contenido no gráfico de los modelos.
- **Relaciones:** Los objetos en un modelo BIM no existen de forma aislada, sino que se vinculan entre sí mediante relaciones espaciales y lógicas que definen cómo interactúan en un entorno 3D. Esto permite comprender mejor el contexto de cada componente dentro del conjunto de la edificación. Por ejemplo, un objeto "Ventana" estaría relacionado con el objeto "Muro" en el que está insertado.

En la práctica, cada vendedor de software posee su propia representación de un modelo BIM. Estos conceptos siempre se encuentran presentes de alguna forma en cada estructura BIM y que es exportable por la mayoría de las herramientas modernas.

2.2 IFC (INDUSTRY FOUNDATION CLASSES):

IFC es un formato de archivo estandarizado y abierto, desarrollado por buildingSMART, cuyo principal objetivo es facilitar la interoperabilidad y el intercambio de información entre diferentes softwares utilizados en la industria de la construcción. Este estándar define un esquema de datos que permite representar y compartir modelos BIM, incluyendo objetos, propiedades y relaciones [buildingSMART s.f.-d].

El esquema IFC se basa en una estructura jerárquica de clases y subclases que abarcan una amplia gama de elementos y conceptos relacionados con la construcción, como geometría, materiales, espacios, equipos, etc. Cada clase define un conjunto de atributos y relaciones que caracterizan a los objetos que pertenecen a ella. Por ejemplo, la clase "IfcWall" representa un muro y contiene atributos como espesor, altura, material, etc.

Una de las características destacadas de IFC es su capacidad para representar no solo la geometría de los objetos, sino también su semántica y comportamiento. Esto significa que además de la forma y dimensiones de un elemento, IFC puede incluir información sobre su función, propiedades técnicas, relaciones con otros objetos y más. Esta riqueza semántica es fundamental para realizar análisis y simulaciones avanzadas, como cálculos estructurales, estudios energéticos o verificaciones normativas.

Al exportar un modelo BIM a formato IFC, la información se organiza según este esquema estandarizado, lo que da paso a que distintas aplicaciones puedan interpretar y utilizar los datos de manera consistente. Esto es fundamental para garantizar la colaboración efectiva entre los diversos actores involucrados en un proyecto de construcción, como arquitectos, ingenieros, contratistas y fabricantes.

Algunos de los principales beneficios de utilizar el formato IFC son:

- **Interoperabilidad:** Permite el intercambio fluido de información entre diferentes softwares BIM, evitando la pérdida de datos y facilitando la colaboración entre disciplinas.
- **Preservación de la información:** Al ser un formato abierto y estandarizado, IFC garantiza que la información del modelo BIM se mantenga accesible a largo plazo, independientemente de los cambios en las aplicaciones utilizadas.
- **Flexibilidad:** El esquema IFC es extensible y permite incorporar información específica de cada proyecto o disciplina, adaptándose a las necesidades particulares de cada caso.
- **Soporte para el ciclo de vida completo:** IFC abarca todas las etapas del ciclo de vida de una construcción, desde el diseño hasta la operación y mantenimiento, lo que permite una gestión integral de la información a lo largo de todo el proceso.

En el contexto de la verificación automatizada de normativas, el uso de IFC es esencial, ya que proporciona una estructura de datos común y estandarizada que facilita el análisis y la evaluación de los modelos BIM. Las reglas y requisitos normativos pueden ser programados para interpretar la información contenida en un archivo IFC, permitiendo una verificación más eficiente y precisa del cumplimiento de las regulaciones aplicables.

Además, IFC permite la creación de subconjuntos de información o "*Model Views*" que extraen solo los datos relevantes para un propósito específico, como la verificación de normativas de seguridad contra incendios o accesibilidad. Esto optimiza el proceso de análisis y reduce la complejidad de los modelos.

En resumen, IFC es un formato de archivo fundamental para garantizar la interoperabilidad, el intercambio efectivo de información y la colaboración en el entorno BIM. Su capacidad para representar la semántica y el comportamiento de los objetos, junto con su flexibilidad y soporte para el ciclo de vida completo de la construcción, lo convierten en un componente clave para la verificación automatizada de normativas en proyectos de construcción.

2.3 LENGUAJE DE DOMINIO ESPECÍFICO (DSL):

Un lenguaje de dominio específico, es un lenguaje de programación diseñado para abordar un problema o dominio particular, para nuestro caso el dominio Arquitectura y Normativas de construcción. A diferencia de un lenguaje de programación de propósito general, que se diseñan para ser aplicables para una amplia gama de problemas, los DSL se centran en un ámbito específico, proporcionando abstracciones y notaciones especializadas que se adaptan a las necesidades y conceptos propios del dominio.

El uso de lenguajes de dominio específico (DSL) en el contexto de la construcción permite definir reglas precisas y específicas que pueden ser fácilmente interpretadas por sistemas automatizados para verificar el cumplimiento de normativas y códigos de construcción. Un DSL proporciona una sintaxis y semántica que se ajusta a las necesidades particulares del dominio de aplicación, facilitando la creación y validación de modelos BIM [Sydora et al., 2020].

Los DSL utilizados para la verificación de normas en la construcción permiten representar propiedades geométricas y no geométricas de los objetos del modelo, definir relaciones entre estos objetos y especificar condiciones lógicas que deben cumplirse para garantizar la conformidad con las normativas vigentes [Dimiyadi et al., 2016].

Además, los lenguajes de dominio específico son esenciales para modelar reglas complejas que implican múltiples condiciones y relaciones entre los elementos del modelo BIM. Estas reglas pueden abarcar desde simples verificaciones de propiedades individuales hasta evaluaciones más complejas que involucran múltiples elementos y sus interacciones dentro del modelo [Zhong et al., 2012].

La principal ventaja de un DSL es que permite expresar soluciones de manera concisa y cercana al lenguaje utilizados por los expertos del dominio, lo que idealmente facilitaría la comunicación y desarrollo de software en ese campo específico. Esto en nuestro dominio permite:

- **Representación de normas de construcción:** Un DSL permite diseñar representación de manera clara y estructurada de las normas y requisitos establecidos en la actual normativa de construcción chilena. Esto implica definir una sintaxis y una semántica que permitan expresar las reglas, restricciones y criterios que deben cumplir los planos arquitectónicos. El DSL actuaría como un puente entre el lenguaje técnico utilizado en las normas y una representación comprensible para un sistema automatizado.
- **Validación y análisis automatizados:** Al tener las normas de construcción expresadas en un DSL, es posible desarrollar herramientas y algoritmos que analicen automáticamente los planos arquitectónicos en busca de incumplimientos o inconsistencias respecto a esas normas. El DSL proporcionaría una estructura clara y formal que facilitaría la implementación de reglas de validación y la detección de errores o desviaciones en los planos.
- **Integración con herramientas de diseño:** Un DSL que se encuentre bien definido podría integrarse con herramientas de diseño, entornos virtuales gubernamentales o programas externos. Siendo que el programa capaz de interpretar el DSL sea un motor que potencie una variedad de utilidades tales como herramientas para crear y visualizar las reglas, implementación web para una gestión virtual de las revisiones, integración con programas de diseño BIM para una revisión a tiempo real, etc.

- **Adaptabilidad a cambios normativos:** Al tener las normas de construcción expresadas en un DSL, resulta más sencillo adaptar el sistema de revisión automatizada a los cambios y actualizaciones que puedan surgir en la normativa chilena. Modificar las reglas y criterios en el DSL sería más directo y menos propenso a errores que actualizar un sistema basado en código de propósito general.
- **Comunicación y colaboración:** Un DSL bien diseñado puede mejorar la comunicación y colaboración entre los diferentes actores involucrados en el proceso de revisión de planos, como arquitectos, ingenieros y revisores. Al utilizar un lenguaje especializado y común, se reduce la ambigüedad y se facilita el entendimiento de los requisitos y criterios establecidos en las normas de construcción.

2.4 REPRESENTACIÓN Y PROCESAMIENTO DE OBJETOS 3D:

La representación y procesamiento de objetos 3D es un aspecto fundamental para diversas aplicaciones tanto en el campo de la arquitectura, como la ingeniería y la construcción. Existen dos formas principales de representar la geometría de los objetos 3D de manera computacional, puede ser mediante objetos triangulados y mediante objetos lineales.

Los objetos triangulados se componen de un conjunto de vértices en el espacio 3D, junto con un conjunto de índices que definen la conectividad entre estos vértices, formando un conjunto de triángulos. Esta representación permite modelar de forma precisa y detallada la superficie de los objetos, capturando con exactitud su geometría. Cada triángulo está definido por tres vértices, y el conjunto de todos los triángulos define la superficie del objeto. Este enfoque es ampliamente utilizado para representar modelos 3D complejos, como los utilizados en aplicaciones de diseño asistido por computadora (CAD), visualización de entornos virtuales y procesamiento de escaneos 3D.

Por otro lado, los objetos lineales se representan mediante un conjunto de vértices en el espacio 3D, junto con un conjunto de pares de índices que definen los segmentos o aristas que conectan dichos vértices. Esta representación es más adecuada para modelar elementos lineales, como tuberías, cables, vigas y otros componentes estructurales. A diferencia de los objetos triangulados, los objetos lineales no capturan la superficie completa del objeto, sino que se enfocan en la representación de sus bordes y aristas.

En IFC, los objetos 3D se describen utilizando tanto mallas trianguladas como segmentos lineales. Esto permite modelar de manera precisa la geometría de los elementos constructivos, como muros, vigas, ventanas y puertas, entre otros y los elementos referenciales, como la línea oficial, eje público, ejes de deslindes, entre otras líneas descritas previamente en las definiciones del OGUC.

Sin embargo, trabajar directamente con la descripción de vértices, aristas y caras de estos objetos 3D plantea desafíos. Es necesario implementar algoritmos eficientes para extraer y procesar las propiedades geométricas relevantes, como la detección de intersecciones, la determinación de la orientación de los objetos, el cálculo de distancias y alturas, y la identificación de relaciones espaciales entre elementos. Estas tareas requieren un manejo adecuado de las estructuras de datos que representan la geometría, así como la aplicación de técnicas computacionales avanzadas para garantizar un procesamiento eficiente y preciso de la información.

CAPÍTULO 3: PROPUESTA DE SOLUCION

3.1 INTRODUCCIÓN:

En este capítulo se presentará la propuesta de solución para la automatización de la verificación del cumplimiento normativo en proyectos arquitectónicos, utilizando tecnologías avanzadas como BIM y un DSL. El objetivo principal es desarrollar un sistema que permitirá la evaluación automática de los proyectos arquitectónicos contra las normativas vigentes, mejorando la eficiencia y precisión en el proceso de revisión.

La propuesta se estructura en varias secciones que detallan desde la metodología de desarrollo empleada, los requisitos del sistema, hasta la arquitectura técnica de la solución. Este enfoque integral asegurará que todos los aspectos del desarrollo del software estén cubiertos, proporcionando una visión clara y completa del sistema.

Primero, se detallarán las prácticas y principios de ingeniería de software aplicados durante el desarrollo del proyecto. Esta sección incluye el ciclo de vida de desarrollo del software, la gestión de configuración y las prácticas de aseguramiento de la calidad.

Seguidamente, se describirá la metodología específica utilizada para el desarrollo del software, incluyendo las fases del desarrollo, los roles y responsabilidades de los miembros del equipo, y la implementación de la metodología seleccionada.

Posteriormente, se especificarán los requisitos funcionales y no funcionales del sistema, detallando las funcionalidades clave que deberá ofrecer el software y los criterios de rendimiento, seguridad y usabilidad que debe cumplir.

La arquitectura técnica del sistema se presentará describiendo la visión general de la arquitectura, los componentes principales del sistema, las interacciones entre estos componentes, y las tecnologías utilizadas. También se incluirá un diagrama de arquitectura que ilustra la estructura del sistema.

En una sección subsecuente, basada en la metodología de Christoph Sydora y Eleni Stroulia [Sydora, 2020], se describe el diseño del lenguaje para la expresión de reglas normativas. Se explicará la sintaxis del lenguaje, cómo se evalúan las reglas basándose en múltiples instancias de objetos relevantes, y se presenta un editor para el lenguaje de reglas.

Para demostrar la sintaxis del lenguaje, se utilizará la normativa de distanciamientos a los deslindes requeridos para construcciones de tipo aislado según el artículo 2.6.3 de la OGUC. Se describen las reglas en lenguaje natural y su transcripción al lenguaje creado, incluyendo un diagrama que ilustra estas reglas.

Esta estructura proporciona una visión integral del desarrollo del software, cubriendo todos los aspectos necesarios para entender cómo se diseñó, desarrolló y se implementó la solución propuesta.

3.2 INGENIERÍA DE SOFTWARE:

Los primeros pasos fue identificar las estructuras que tendrían los modelos 3D, diferenciando entre objetos lineales y triangulados con tal de crear estructuras concretas diferenciadoras. Los objetos lineales se representaron como una tupla que contenía una lista de vértices y una lista de índices, mientras que los objetos triangulados se representaron con una lista de índices y una lista de coordenadas. En los objetos lineales, cada vértice correspondía a tres coordenadas en una tupla (x, y, z), mientras que, en los objetos triangulados, cada conjunto de tres coordenadas correspondía a un vértice en la lista de coordenadas.

El siguiente paso fue diseñar la estructura básica de un DSL para expresar las primeras reglas normativas en un archivo de texto. Se inició el proceso de creación de una estructura de lectura para dicho archivo, desarrollando metodologías para revisar la gramática del archivo utilizando expresiones regulares (regex).

Se implementó una estructura de pilas para capturar los comandos según aparecían en el archivo de texto. Con esta pila de comandos, se comenzó a desarrollar la función de construcción, creando función por función a medida que se iban encontrando en el archivo de ejemplo.

Una vez creadas todas las funciones básicas, se verificó la capacidad del sistema para construir elementos 3D utilizando estas funciones, aplicando un caso práctico de creación de rasantes. Esto permitió validar la funcionalidad del sistema y hacer ajustes necesarios.

El desarrollo continuó con la implementación de la función REGLA, estableciendo una metodología para crear las clases regla, cláusula y nodo. Se desarrollaron los métodos para cada clase, y se implementaron funciones para evaluar correctamente una regla.

A medida que surgían nuevas necesidades, se crearon funciones adicionales para soportar las reglas definidas. Se generaron nuevos archivos de texto con las reglas que finalmente se utilizarían para la evaluación.

Se realizaron pruebas continuas para ajustar todas las funciones creadas, asegurando la robustez del sistema. Estas pruebas incluyeron la revisión de varios proyectos arquitectónicos para poner a prueba el sistema en escenarios reales.

Durante la fase de pruebas, se identificaron y corrigieron errores encontrados en nuevos proyectos arquitectónicos, ajustando las funciones existentes según fuera necesario. Finalmente, se llevó a cabo una prueba de conceptos con un arquitecto profesional, validando el sistema en un entorno profesional y obteniendo retroalimentación para mejoras adicionales.

El desarrollo de este software fue un proceso iterativo y centrado en la resolución de problemas específicos, adaptándose continuamente a las necesidades emergentes y refinando las funcionalidades del sistema a través de pruebas y ajustes continuos.

3.2.1 PRÁCTICAS Y PRINCIPIOS DE INGENIERÍA DE SOFTWARE

Ciclo de Vida del Desarrollo:

- **Modelo Iterativo:** El desarrollo se realizó en ciclos cortos enfocados en objetivos específicos, permitiendo ajustes continuos y mejoras incrementales.
- **Etapas:** Incluyó la identificación de requisitos, diseño, implementación, pruebas y mantenimiento continuo.

Gestión de Configuración:

- **Control de Versiones:** Se utilizó un sistema de control de versiones manual, realizando respaldos en la nube después de completar cada objetivo importante.
- **Respaldo en la Nube:** Cada vez que se alcanzaba un objetivo del desarrollo, se realizaba un respaldo en una plataforma de almacenamiento en la nube para asegurar la disponibilidad y protección de los datos.

Aseguramiento de la Calidad:

- **Pruebas Unitarias:** Implementación de pruebas unitarias para verificar la funcionalidad de componentes individuales del software.
- **Pruebas de Integración:** Validación de la interacción entre diferentes módulos del sistema.
- **Pruebas de Sistema:** Evaluación del sistema completo en escenarios reales para asegurar su funcionamiento correcto.
- **Pruebas de Aceptación:** Realización de pruebas de aceptación con un arquitecto profesional para validar el sistema en un entorno profesional.

Se resumen todo lo anterior en la **Tabla 2**:

Práctica de Ingeniería de Software	Descripción
Ciclo de Vida del Desarrollo	Enfoque iterativo y adaptativo con ciclos cortos de desarrollo.
Control de Versiones	Respaldo manual en la nube después de completar cada objetivo importante.
Respaldo en la Nube	Almacenamiento de datos en la nube para asegurar disponibilidad y protección.
Pruebas Unitarias	Implementación de pruebas unitarias para verificar la funcionalidad de componentes individuales.
Pruebas de Integración	Validación de la interacción entre diferentes módulos del sistema.
Pruebas de Sistema	Evaluación del sistema completo en escenarios reales.
Pruebas de Aceptación	Realización de pruebas de aceptación con un arquitecto profesional.

Tabla 2: Resumen practicas Ing. Software

Fuente: Elaboración Propia

3.3 METODOLOGÍA DE DESARROLLO:

La metodología de desarrollo aplicada en este proyecto se caracterizó por un enfoque iterativo y adaptativo, sin una planificación previa estructurada. Este enfoque permitió una mayor flexibilidad y adaptabilidad, permitiendo ajustarse a las necesidades emergentes y objetivos específicos del proyecto en cada etapa. A continuación, se detallan los aspectos clave de la metodología de desarrollo utilizada.

3.3.1 ENFOQUE ITERATIVO Y ADAPTATIVO:

El desarrollo del software se realizó en ciclos cortos y repetitivos, permitiendo incorporar nuevas funcionalidades y realizar ajustes continuos según se identificaban las necesidades. Este enfoque permitió avanzar de manera progresiva, asegurando que cada iteración aportara valor al sistema en desarrollo.

3.3.2 FASES DEL CICLO DE VIDA DEL DESARROLLO:

- **Identificación de Requisitos:** En cada iteración, se identificaban los requisitos específicos necesarios para alcanzar los objetivos definidos. Estos requisitos se basaban en las metas planteadas, como la capacidad de leer archivos IFC, crear diccionarios de clases IFC con geometría 3D, y desarrollar funciones específicas para el DSL.

- **Diseño:** Una vez identificados los requisitos, se procedía a diseñar la solución técnica correspondiente. Este diseño incluía la definición de estructuras de datos, algoritmos y funciones necesarias para cumplir con los requisitos planteados.
- **Implementación:** Con el diseño definido, se procedía a la implementación del código correspondiente. Esta etapa involucraba la programación de las funciones y métodos necesarios, así como la integración de nuevas funcionalidades en el sistema existente.
- **Pruebas:** Después de cada implementación, se realizaban pruebas para verificar la funcionalidad y correctitud del sistema. Estas pruebas incluían pruebas unitarias, pruebas de integración y pruebas de sistema.
- **Mantenimiento y Ajustes:** Basado en los resultados de las pruebas y la retroalimentación obtenida, se realizaban los ajustes necesarios para corregir errores y mejorar el rendimiento del sistema. Este ciclo se repetía hasta que se alcanzaran los objetivos del proyecto.

3.3.3 Roles y Responsabilidades:

Dado que el desarrollo del proyecto fue realizado por una sola persona, todas las responsabilidades de identificación de requisitos, diseño, implementación, pruebas y mantenimiento recayeron en el desarrollador. Esto permitió una toma de decisiones ágil y una implementación rápida de las funcionalidades necesarias.

3.3.4 TÉCNICAS Y HERRAMIENTAS UTILIZADAS:

- **Expresiones Regulares (Regex):** Utilizadas para revisar la gramática de los archivos de reglas y asegurar su correcta interpretación.
- **Estructuras de Pilas:** Implementadas para capturar y gestionar los comandos según aparecían en el archivo de texto del DSL.
- **Control de Versiones:** Se realizó un respaldo manual en la nube cada vez que se alcanzaba un objetivo importante del desarrollo, asegurando la protección y disponibilidad de los datos.

3.3.5 LIBRERÍAS Y TECNOLOGÍAS ESPECÍFICAS:

- **NumPy:** Utilizado para operaciones matemáticas y de álgebra lineal, especialmente en el manejo y manipulación de geometrías 3D. NumPy facilitó el cálculo de vectores normales, distancias y otras operaciones geométricas necesarias para la evaluación de reglas.
- **IFC-OpenShell:** Una librería de Python para trabajar con archivos IFC. IFC-OpenShell permitió la lectura y manipulación de archivos IFC, extrayendo información geométrica y propiedades de los elementos del modelo BIM.
- **Matplotlib:** Empleada para la visualización de datos y geometrías. Matplotlib facilitó la creación de gráficos y diagramas que ayudaron a visualizar los resultados de las evaluaciones de reglas.

- **Regex:** Utilizada para la validación y parsing de archivos de texto, asegurando que las reglas definidas en el DSL fueran interpretadas correctamente.

3.3.6 PRUEBAS Y VALIDACIÓN:

Se implementaron diversas técnicas de pruebas para asegurar la calidad del software:

- **Pruebas Unitarias:** Verificación de la funcionalidad de componentes individuales del software.
- **Pruebas de Integración:** Validación de la interacción entre diferentes módulos del sistema.
- **Pruebas de Sistema:** Evaluación del sistema completo en escenarios reales para asegurar su correcto funcionamiento.
- **Pruebas de Aceptación:** Realización de pruebas de aceptación con un arquitecto profesional para validar el sistema en un entorno profesional.

3.4 REQUISITOS DEL SISTEMA:

La definición clara y precisa de los requisitos del sistema es esencial para garantizar que el software desarrollado cumpla con los objetivos y necesidades del proyecto. En esta sección, se describen los requisitos funcionales y no funcionales del sistema de verificación automatizada del cumplimiento normativo en proyectos arquitectónicos.

3.4.1 REQUISITOS FUNCIONALES

Identificador	RF-01
Título	Lectura de Archivos IFC
Descripción	El sistema debe ser capaz de leer y parsear archivos en formato IFC, extrayendo la información relevante de los modelos BIM.
Justificación	La lectura de archivos IFC es fundamental para obtener los datos necesarios para la evaluación normativa.
Criterios de Aceptación	• El sistema deberá poder abrir y leer archivos IFC 4. • La información geométrica y de propiedades deberá ser extraída correctamente.
Prioridad	Alta
Dependencias	Ninguna

Tabla 3: Requisito Funcional-01

Fuente: Elaboración propia

Identificador:	RF-02
Título:	Identificación y Almacenamiento de Clases IFC con Geometría 3D.
Descripción:	El sistema deberá crear un diccionario que identifique y almacene las clases IFC que contienen geometría 3D.
Justificación:	La capacidad de identificar y almacenar clases con geometría 3D es crucial para la evaluación precisa de los modelos BIM, facilitando la verificación de cumplimiento normativo.
Criterios de Aceptación:	• El diccionario deberá almacenar correctamente las clases IFC con geometría 3D. • La información almacenada deberá ser accesible para su posterior procesamiento
Prioridad	Alta
Dependencias	RF-01

Tabla 4: Requisito Funcional-02

Fuente: Elaboración propia

Identificador:	RF-03
Título:	Estructuración de Modelos 3D
Descripción:	El sistema debe ser capaz de estructurar los modelos 3D, diferenciando entre objetos lineales y triangulados.
Justificación:	Diferenciar entre objetos lineales y triangulados permitirá una mejor manipulación y evaluación de las geometrías 3D dentro del sistema.
Criterios de Aceptación:	• Los objetos lineales deben ser representados como una tupla con listas de vértices e índices. • Los objetos triangulados deben ser representados con listas de índices y vértices.
Prioridad	Alta
Dependencias	RF-02

Tabla 5: Requisito Funcional-03

Fuente: Elaboración propia

Identificador:	RF-04
Título:	Lectura de DSL para Reglas Normativas
Descripción:	El sistema deberá permitir la lectura de archivos de texto que contengan las reglas normativas en un lenguaje de dominio específico (DSL) y ser capaz de leer y parsear estos archivos, validando su gramática utilizando expresiones regulares.
Justificación:	La creación y lectura de un DSL permitirá una definición clara y precisa de las reglas normativas, facilitando su evaluación automática.
Criterios de Aceptación:	• El sistema deberá permitir la lectura de archivos de texto con reglas en DSL. • Deberá validar la gramática de estos archivos utilizando expresiones regulares.
Prioridad	Alta
Dependencias	RF-01

Tabla 6: Requisito Funcional-04

Fuente: Elaboración propia

Identificador:	RF-05
Título:	Implementación de Estructura de Pilas para Comandos DSL
Descripción:	El sistema implementará una estructura de pilas que almacenará los comandos y respetará las indentaciones y lógicas de la sintaxis del DSL.
Justificación:	Utilizar una estructura de pilas asegurará que los comandos se procesen en el orden correcto y que se mantengan las relaciones lógicas y de indentación entre ellos.
Criterios de Aceptación:	• El sistema deberá ser capaz de almacenar y procesar comandos DSL en una estructura de pilas. • Deberá respetar las indentaciones y lógicas de la sintaxis del DSL durante el procesamiento.
Prioridad	Alta
Dependencias	RF-04

Tabla 7: Requisito Funcional-05

Fuente: Elaboración propia

Identificador:	RF-06
Título:	Evaluación de Reglas Normativas
Descripción:	El sistema deberá implementar una función de evaluación de reglas que permitirá verificar el cumplimiento normativo de los elementos del modelo BIM, incluyendo la capacidad de definir y evaluar reglas complejas mediante una estructura de clases (regla, cláusula y nodo).
Justificación:	La evaluación automática de reglas normativas es esencial para asegurar que los proyectos arquitectónicos cumplan con las normativas vigentes de manera eficiente y precisa.
Criterios de Aceptación:	• El sistema deberá evaluar reglas normativas definidas en el DSL. • Deberá permitir la definición de reglas complejas utilizando una estructura de clases.
Prioridad	Alta
Dependencias	RF-05

Tabla 8: Requisito Funcional-06

Fuente: Elaboración propia

Identificador:	RF-07
Título:	Visualización de Resultados.
Descripción:	El sistema deberá proporcionar una visualización de los resultados de la evaluación, permitiendo identificar fácilmente las áreas que cumplen o no con las normativas.
Justificación:	La visualización de resultados facilitará la interpretación y toma de decisiones por parte de los usuarios, mejorando la usabilidad del sistema.
Criterios de Aceptación:	• El sistema deberá visualizar claramente los resultados de la evaluación de las reglas. • Deberá permitir la identificación de áreas que no cumplen con las normativas.
Prioridad	Media
Dependencias	RF-06

Tabla 9: Requisito Funcional-07

Fuente: Elaboración propia

3.4.2 REQUISITOS NO FUNCIONALES

Identificador:	RNF-01
Título:	Rendimiento
Descripción:	El sistema deberá ser capaz de procesar archivos IFC y evaluar reglas normativas en un tiempo razonable, preferiblemente en menos de 15 segundos para archivos de tamaño medio.
Justificación:	Un buen rendimiento es crucial para la aceptación del sistema por parte de los usuarios finales, quienes esperan resultados rápidos y precisos.
Criterios de Aceptación:	El procesamiento de un archivo IFC de tamaño medio debe completarse en menos de 15 segundos en un entorno de prueba estándar.
Prioridad	Alta
Dependencias	RF-01, RF-06

Tabla 10: Requisito No Funcional-01**Fuente:** Elaboración propia

Identificador:	RNF-02
Título:	Escalabilidad
Descripción:	El sistema deberá ser escalable, permitiendo la adición de nuevas reglas normativas y la capacidad de manejar modelos BIM de mayor complejidad y tamaño.
Justificación:	La escalabilidad es importante para asegurar que el sistema podrá evolucionar y seguir siendo útil a medida que aumentan los requisitos y la complejidad de los proyectos.
Criterios de Aceptación:	- El sistema debe poder incorporar nuevas reglas normativas sin necesidad de reestructuraciones significativas. - Debe manejar eficientemente modelos BIM de gran tamaño y complejidad.
Prioridad	Media
Dependencias	RF-01, RF-02

Tabla 11: Requisito No Funcional-02**Fuente:** Elaboración Propia

Identificador:	RNF-03
Título:	Compatibilidad
Descripción:	El sistema debe ser compatible con las versiones IFC 4, asegurando la interoperabilidad con diverso software de diseño arquitectónico.
Justificación:	La compatibilidad con múltiples versiones de IFC garantizará que el sistema pueda ser utilizado en una amplia variedad de entornos y proyectos.
Criterios de Aceptación:	El sistema debe poder leer y procesar archivos en formato y IFC 4 sin problemas de compatibilidad.
Prioridad	Alta
Dependencias	RF-01

Tabla 12: Requisito No Funcional-03**Fuente:** Elaboración propia

Identificador:	RNF-04
Título:	Usabilidad
Descripción:	El sistema deberá ser fácil de usar, con una interfaz intuitiva que permitirá a los usuarios definir reglas normativas y evaluar modelos BIM sin necesidad de conocimientos técnicos
Justificación:	Una alta usabilidad es esencial para asegurar que el sistema sea adoptado y utilizado efectivamente por usuarios con diferentes niveles de habilidad técnica.
Criterios de Aceptación:	- La interfaz del usuario debe ser intuitiva y fácil de navegar. - Los usuarios deberán poder definir y evaluar reglas sin necesidad de conocimientos técnicos avanzados.
Prioridad	Baja
Dependencias	RF-01, RF-04

Tabla 13: Requisito No Funcional-04**Fuente:** Elaboración propia

3.5 ARQUITECTURA TÉCNICA:

La arquitectura técnica del sistema de verificación automatizada del cumplimiento normativo en proyectos arquitectónicos se diseñó para garantizar la eficiencia, escalabilidad y mantenibilidad del software. A continuación, se describe en detalle la arquitectura del sistema, destacando sus componentes principales y las interacciones entre ellos.

3.5.1 VISIÓN GENERAL DE LA ARQUITECTURA

El sistema se compone de varios componentes clave que interactúan entre sí para proporcionar una funcionalidad completa de lectura de archivos IFC, evaluación de reglas normativas y visualización de resultados. La arquitectura general del sistema puede dividirse en las siguientes capas:

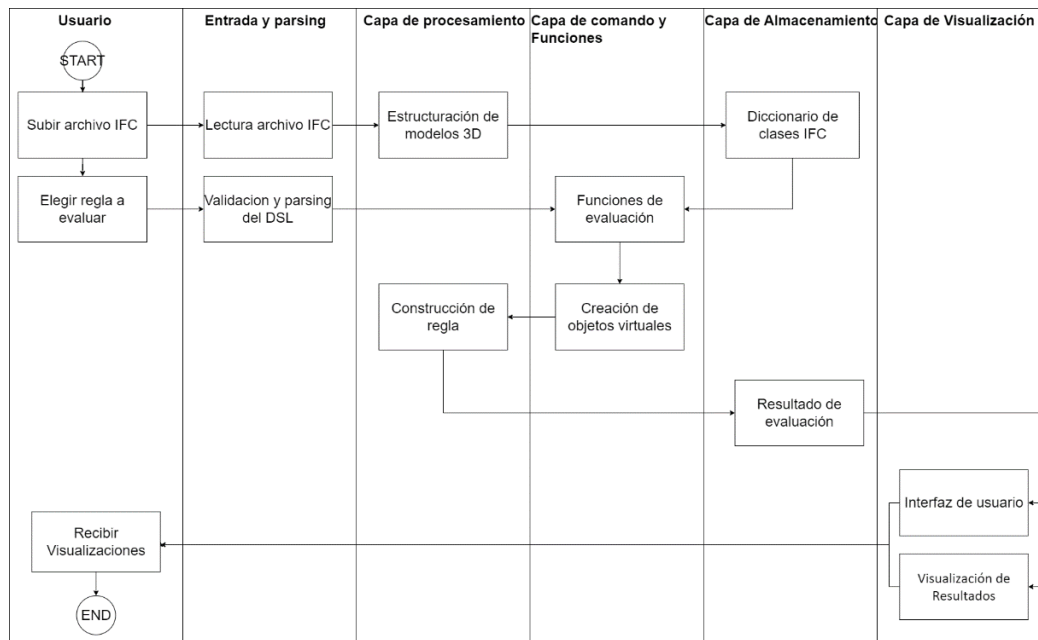
1. Capa de Entrada y Parsing:
 - Lectura de Archivos IFC: Componente responsable de cargar y parsear archivos IFC, extrayendo la información relevante de los modelos BIM.
 - Validación y Parsing del DSL: Componente que lee y parsea archivos de texto que contienen las reglas normativas en un lenguaje de dominio específico (DSL).
2. Capa de Procesamiento:
 - Estructuración de Modelos 3D: Módulo que estructura los modelos 3D, diferenciando entre objetos lineales y triangulados.
 - Construcción de Reglas: Módulo que implementa la lógica de evaluación de reglas normativas utilizando estructuras de clases (regla, cláusula y nodo) y una estructura de pilas para procesar los comandos DSL.
3. Capa de Comandos y Funciones:
 - Creación de Objetos Virtuales: Componente que permite la creación de objetos virtuales en el entorno BIM, como elementos estructurales y arquitectónicos.
 - Funciones de Evaluación: Conjunto de funciones específicas para evaluar reglas y realizar cálculos necesarios para la verificación normativa.
4. Capa de Almacenamiento:
 - Diccionario de Clases IFC: Almacena las clases IFC que contienen geometría 3D para facilitar su posterior procesamiento.
 - Resultados de Evaluación: Almacena los resultados de la evaluación de las reglas normativas.
5. Capa de Visualización:
 - Interfaz de Usuario: Componente que proporciona una interfaz intuitiva para la visualización de los resultados de la evaluación y la interacción con el sistema.
 - Visualización de Resultados: Módulo que muestra los resultados de la evaluación de las reglas, destacando las áreas que cumplen o no con las normativas.

3.5.2 COMPONENTES DEL SISTEMA

1. **Lectura de Archivos IFC** Este componente utiliza la librería IFC-OpenShell para leer y parsear archivos IFC. Extrae información geométrica y de propiedades, almacenándola en el diccionario de clases IFC.

2. **Validación y Parsing del DSL** Utiliza expresiones regulares (regex) para validar y parsear archivos de texto que contienen reglas normativas en DSL. Este componente asegura que las reglas estén bien formadas y listas para su evaluación.
3. **Estructuración de Modelos 3D** Diferencia entre objetos lineales y triangulados. Los objetos lineales se representan como una tupla con listas de vértices e índices, mientras que los objetos triangulados se representan con listas de índices y vértices.
4. **Creación de Reglas** Implementa la lógica de evaluación de reglas normativas. Utiliza estructuras de clases (regla, cláusula y nodo) y una estructura de pilas para procesar comandos DSL, respetando las indentaciones y lógicas de la sintaxis del DSL.
5. **Creación de Objetos Virtuales** Este componente permite la creación de objetos virtuales dentro del entorno BIM, como muros, ventanas, y otros elementos arquitectónicos y estructurales. Las funciones específicas para crear estos objetos se definen y gestionan en este módulo.
6. **Funciones de Evaluación** Incluye un conjunto de funciones específicas que realizan cálculos necesarios para evaluar reglas normativas, como el cálculo de distancias, verificación de ángulos, y otros parámetros críticos.
7. **Diccionario de Clases IFC** Almacena las clases IFC que contienen geometría 3D. Este diccionario facilita el acceso y procesamiento de la información geométrica extraída de los archivos IFC.
8. **Resultados de Evaluación** Almacena los resultados de la evaluación de las reglas normativas. Este componente permite realizar análisis posteriores y generar informes de cumplimiento.
9. **Interfaz de Usuario** Proporciona una interfaz intuitiva para que los usuarios interactúen con el sistema. Permite cargar archivos IFC, definir reglas normativas y visualizar los resultados de la evaluación.
10. **Visualización de Resultados** Muestra los resultados de la evaluación de las reglas, permitiendo a los usuarios identificar fácilmente las áreas que cumplen o no con las normativas. Utiliza gráficos y diagramas para representar la información de manera clara y comprensible

Se puede mostrar todos los componentes y capas como interactúan entre si desde que el usuario desea evaluar una regla de un proyecto hasta recibir las visualizaciones en la **Figura 2**.

**Figura 2:** Diagrama arquitectura básica**Fuente:** Elaboración propia

3.6 REPRESENTACIÓN DE REGLAS:

En base a la metodología realizada por Christoph Sydora y Eleni Stroulia [Sydora20] el lenguaje para las reglamentaciones está diseñado para expresar reglas básicas sobre:

- Propiedades geométricas y no geométricas de objetos reales o virtuales de elementos IFC.
- Relaciones geométricas entre elementos.
- Reglas complejas que son composiciones lógicas de los dos tipos anteriores de reglas básicas.

Dado que muchas normas se basan intrínsecamente en la lógica operacional, el lenguaje creado deriva gran parte en el razonamiento lógico matemático. Es fácil ver la similitud entre una afirmación “para cada x en los números reales” en matemáticas con “para cada ventana...” en las normas de construcción. También se dará uso a lógica booleana para la composición básica de las comprobaciones. En las secciones siguientes se describe la sintaxis de la regla, como se evalúa una regla basándose en múltiples instancias de los objetos de relevantes, y una implementación de un editor para el lenguaje de reglas.

3.7 SINTAXIS DE REGLAS:

Para demostrar la sintaxis del lenguaje presentado se usará como ejemplo la normativa de distanciamientos a los deslindes requeridos para las construcciones de tipo aislado según lo descrito en el artículo 2.6.3 OGUC [MINVU, 2016].

Los distanciamientos medidos en cada una de las alturas de la edificación se señalan en la siguiente tabla:

Altura de la edificación	Distanciamiento	
	Fachada con vano	Fachada sin vano
Hasta 3.5 m	3.0 m	1.4 m
Sobre 3.5 m y hasta 7.0 m	3.0 m	2.5 m
Sobre 7.0 m	4.0 m	4.0 m

Tabla 14: Distanciamientos hacia deslindes
Fuente: Artículo 2.6.3 OGUC [MINVU, 2016].

La **Figura 3** explicita visualmente estas reglas:

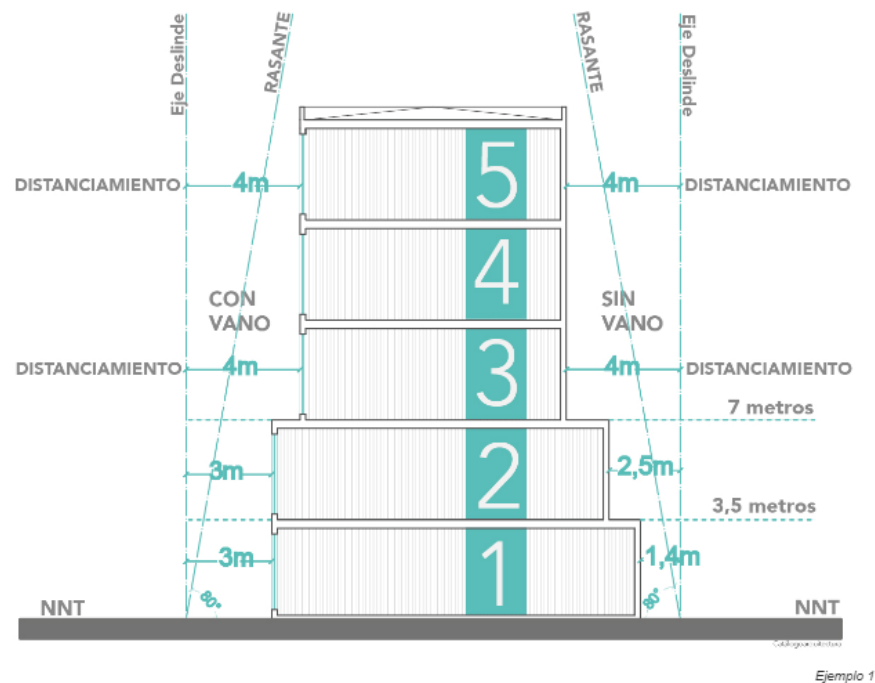


Figura 3: Ejemplo distanciamientos y rasantes
Fuente: OGUC Ilustrada [Catalogo Arquitectura SpA, 2020]

Su transcripción en lenguaje natural sería “para cada elemento de la edificación si su altura es inferior a 3.5 [m] y tiene vano debe distanciarse a más 3.0 [m] del eje del deslinde, si no tiene vano debe distanciarse a más 1.4 [m] del eje del deslinde. Si la altura del elemento es superior a 3.5 [m] y hasta 7.0 [m] y tiene vano debe distanciarse a más de 3.0 [m] del eje del deslinde. Si no tiene vano debe distanciarse a más de 2.5 [m] del eje del deslinde. Si la altura del elemento es superior a 7.0 [m] y tiene vano entonces debe distanciarse a más de 4.0 [m] del eje del deslinde, si no tiene vano se debe distanciar a más de 4,0 [m] del eje.”

Una regla se debe descomponer en los elementos relevantes a evaluar, la relación entre dichos elementos y cláusulas lógicas que describen la regla.

En este caso los elementos relevantes están descritos en la sentencia “cada elemento de la edificación”, lo cual, usualmente describe los elementos de obra gruesa tales como muros, tejados, pisos, etc. Y también el elemento “eje del deslinde”, el que está descrito por un plano recto a la línea del deslinde o, dicho de otro modo, límite del terreno. Por otro lado, se sabe que todo elemento de la edificación debe cumplir el distanciamiento descrito con todos los ejes los deslindes.

Las cláusulas lógicas son las condiciones individuales que se evalúan en la regla. Cada cláusula debe estar compuesta por una expresión lógica que compara dos atributos o propiedades de los objetos utilizando operadores lógicos. Además, el orden y razonamiento de las cláusulas deben respetar el de una operación lógica proposicional. Para el ejemplo anterior se pueden identificar 6 segmentos de condiciones lógicas, tres que relacionan la condición de la altura de los elementos versus la distancia al eje del deslinde si poseen vano y otros tres que siguen la misma lógica, pero sin vano. Las **Tabla 15** y **Tabla 16** describe las condiciones para elementos entre 0.0 [m] y 3.5 [m] con y sin vano respectivamente. Las **Tabla 17** y **Tabla 18** describen las condiciones para elementos entre 3.5 [m] y 7.0 [m] con y sin vano respectivamente. Las tablas **Tabla 19** y **Tabla 20** describen las condiciones para elementos de 7.0 [m] de altura con y sin vano respectivamente.

Clausula	Atributo 1	Operador	Atributo2	Relación
1	altura_desde_terreno_max(ELEMENTO)	<=	3.5 [m]	AND
2	fachada_con_vanos(ELEMENTO)	==	True	AND
3	distancia_entre(ELEMENTO,EJE DESLINDE)	>=	3 [m]	

Tabla 15: Primera sección regla Distanciamiento

Fuente: Elaboración propia

Clausula	Atributo 1	Operador	Atributo2	Relación
4	altura_desde_terreno_max(ELEMENTO)	<=	3.5 [m]	AND
5	fachada_con_vanos(ELEMENTO)	==	False	AND
6	distancia_entre(ELEMENTO, EJE DESLINDE)	>=	1.4 [m]	

Tabla 16: Segunda sección regla Distanciamiento

Fuente: Elaboración propia

Clausula	Atributo 1	Operador	Atributo2	Relación
7	altura_desde_terreno_max(ELEMENTO)	>=	3.5 [m]	AND
8	altura_desde_terreno_max(ELEMENTO)	<=	7.0 [m]	AND
9	fachada_con_vanos(ELEMENTO)	==	True	AND
10	distancia_entre(ELEMENTO, EJE DESLINDE)	>=	3.0 [m]	

Tabla 17: Tercera sección regla Distanciamiento

Fuente: Elaboración propia

Clausula	Atributo 1	Operador	Atributo2	Relación
11	altura_desde_terreno_max(ELEMENTO)	>=	3.5 [m]	AND
12	altura_desde_terreno_max(ELEMENTO)	<=	7.0 [m]	AND
13	fachada_con_vanos(ELEMENTO)	==	False	AND
14	distancia_entre(ELEMENTO, EJE DESLINDE)	>=	2.5 [m]	

Tabla 18: Cuarta sección regla Distanciamiento

Fuente: Elaboración propia

Clausula	Atributo 1	Operador	Atributo2	Relación
15	altura_desde_terreno_max(ELEMENTO)	>=	7.0 [m]	AND
16	fachada_con_vanos(ELEMENTO)	==	True	AND
17	distancia_entre(ELEMENTO, EJE DESLINDE)	>=	4.0 [m]	

Tabla 19: Quinta sección regla Distanciamiento

Fuente: Elaboración propia

Clausula	Atributo 1	Operador	Atributo2	Relación
18	altura_desde_terreno_max(ELEMENTO)	>=	7.0 [m]	AND
19	fachada_con_vanos(ELEMENTO)	==	False	AND
20	distancia_entre(ELEMENTO, EJE DESLINDE)	>=	4.0 [m]	

Tabla 20: Sexta sección regla Distanciamiento

Fuente: Elaboración propia

Así cada sección evalúa independientemente sus respectivas cláusulas y se dirá que un elemento de la edificación cumple la norma si al evaluarlo es verdadera en al menos una de las secciones anteriormente descritas. Esto pues la relación entre secciones describe los tramos de alturas por los que se analizan y si posee vano o no, así que de cumplirse una sección de cláusulas se demuestra que el elemento se encuentra en el tramo de altura de la sección con la opción de vano o sin vano y que está correctamente distanciado entre el elemento y el eje del deslinde. En caso de que para un elemento es falso en las 6 secciones, entonces el elemento no cumple los requisitos de la normativa y se considera que el proyecto no cumple la regla de distanciamiento a deslindes.

Se dice entonces que el resultado de un elemento está descrito por la formula:

Sección 1 **OR** Sección 2 **OR** Sección 3 **OR** Sección 4 **OR** Sección 5 **OR** Sección 6 **=Resultado (1)**

Y a su vez una sección está descrita por las cláusulas y relaciones entre ellas dadas en las tablas anteriores. Luego de haber analizado cada elemento del proyecto versus los planos de los deslindes, se debe relacionar los resultados de cada elemento con cada elemento. Por ejemplo, si estamos revisando una casa de cuatro muros, entonces hemos realizado 4 evaluaciones sobre la fórmula (1), el cual ahora es necesario realizar la relación entre las evaluaciones de los elementos, donde para este caso todo elemento del proyecto se relaciona con todo plano de deslinde. Por lo que para obtener el resultado positivo del proyecto debemos asegurarnos de que todo resultado de la evaluación sea verdadero, para cualquier otro caso el proyecto incumple la norma.

Luego de entender lo anterior se puede describir la normativa de distanciamientos a los ejes de deslindes en nuestro lenguaje como:

ALGORITMO 1: Reglamentación distanciamiento a deslindes

1	REGLA:
2	TODO MUROS, PLANO_DESLINDES
3	
4	EN
5	ALTURA_DESDE_TERRENO(MUROS) <= 3.5 AND
6	FACHADA_CON_VANO(MUROS) == TRUE AND
7	DISTANCIA_PLANO(MUROS, PLANO_DESLINDES) >= 3
8	OR
9	ALTURA_DESDE_TERRENO(MUROS) <= 3.5 AND
10	FACHADA_CON_VANO(MUROS) == FALSE AND
11	DISTANCIA_PLANO(MUROS, PLANO_DESLINDES) >= 1.4
12	OR
13	ALTURA_DESDE_TERRENO(MUROS) >= 3.5 AND
14	ALTURA_DESDE_TERRENO(MUROS) <= 7 AND
15	FACHADA_CON_VANO(MUROS) == TRUE AND
16	DISTANCIA_PLANO(MUROS, PLANO_DESLINDES) >= 3
17	OR
18	ALTURA_DESDE_TERRENO(MUROS) >= 3.5 AND
19	ALTURA_DESDE_TERRENO(MUROS) <= 7 AND
20	FACHADA_CON_VANO(MUROS) == FALSE AND
21	DISTANCIA_PLANO(MUROS, PLANO_DESLINDES) >= 2.5
22	OR
23	ALTURA_DESDE_TERRENO(MUROS) >= 7.0 AND
24	FACHADA_CON_VANO(MUROS) == TRUE AND
25	DISTANCIA_PLANO(MUROS, PLANO_DESLINDES) >= 4.0
26	OR
27	ALTURA_DESDE_TERRENO(MUROS) >= 7.0 AND
28	FACHADA_CON_VANO(MUROS) == TRUE AND
29	DISTANCIA_PLANO(MUROS, PLANO_DESLINDES) >= 4.0
30	TERMINAR

3.8 CHEQUEO DE PROPIEDADES:

Cada elemento que tenga una representación tridimensional en un proyecto arquitectónico es almacenado y clasificado según el formato IFC, de modo que se puede acceder a sus propiedades definidas al momento de realizar el proyecto. Estas pueden ser de tipo numérico, string o booleano. De cualquier modo, la mayoría de las propiedades geométricas se encuentran definidas implícitamente, por lo que es necesario ofrecer funciones que permitan calcular dichas propiedades para ser analizadas en el chequeo de relaciones. Las utilizadas en el prototipo para la revisión de las 4 reglas descritas en el alcance se corresponde con la **Tabla 21**.

Nombre	Descripción
Fachada_con_vano	Retorna un booleano si el objeto tiene un vano en cualquiera de sus caras.
Max_altura	Retorna el valor numérico del punto más alto respecto al centro referencial del modelo.

Tabla 21: Lista de propiedades geométricas

Fuente: Elaboración propia

3.9 CHEQUEO DE RELACIONES:

Para el caso de las relaciones entre objetos en este lenguaje, una relación es en palabras simples, una instancia del resultado del llamado de las funciones descritas en la **Tabla 22** que relaciona dos o más elementos. Por tanto, de forma similar a las propiedades de objeto, si una regla comprueba una propiedad de relación, tal como «Altura_desde», se llama a la función: «Altura_desde(obj1, obj2 cara_obj1, cara_obj2)», pasando ambos objetos como parámetros de la función. Ejemplos de instancias que relacionan objetos es la distancia entre dos objetos o si los objetos se sobreponen. A continuación, el valor puede compararse con los valores de comprobación de la cláusula correspondiente.

Debido a que algunas de las propiedades de relación son direccionales, el orden o dirección de los objetos en la relación es crítico, por ejemplo, según la propiedad «Altura_desde», para obtener una altura implica que el objeto 2 está sobre el objeto 1, de modo que no necesariamente el objeto 1 está encima del objeto 2. Por lo tanto, existirían dos relaciones entre los objetos 1 y 2; una entre el objeto1 y el objeto2, y otra entre el objeto2 y el objeto1, que podrían contener propiedades y valores diferentes. La denominación explícita de las funciones es importante para ayudar a determinar el orden lo mejor posible.

Nombre	Descripción
Linea_cara	Recibe un objeto 3D y una cara (superior o inferior), y retorna una lista de objetos 3D lineales que representan los bordes de esa cara.
Linea_es_sobrepuesta	Recibe dos listas de objetos 3D lineales y un delta, y retorna una lista de los objetos 3D lineales que se superponen.
Linea_distancia	Recibe dos objetos 3D lineales y calcula la menor distancia paralela al plano XY entre ellos, retornando dicha distancia.
Plano_es_sobrepuesto	Recibe dos listas de objetos 3D y umbrales opcionales, y retorna True si alguno de los objetos se superpone.
Altura_desde	Recibe dos objetos 3D y opciones para considerar solo caras superiores, inferiores o cualquiera, y retorna la altura máxima entre los dos objetos.

Tabla 22: Lista de relaciones geométricas

Fuente: Elaboración propia

3.10 CHEQUEO DE COMPOSICIÓN LÓGICA:

El DSL desarrollado en este trabajo permite la definición de reglas complejas que se aplican a los objetos del modelo de construcción. Dichas reglas se componen de un conjunto de cláusulas que evalúan las propiedades y relaciones de los objetos, sin embargo, la estructura lógica que rige la composición de estas cláusulas es fundamental para poder interpretar correctamente el significado y el resultado de la aplicación de una regla.

En el DSL, la estructura lógica de las reglas se representa mediante un árbol de relaciones, donde los nodos pueden ser de dos tipos: nodos de cláusula, que representan una condición individual a evaluar, y nodos de operador lógico, que representan la relación lógica entre las cláusulas (como AND, OR u otras).

Durante la ejecución de una regla, el programa primero construye el árbol de relaciones lógicas a partir de la definición de ésta. La metodología para crear el árbol de relaciones lógicas es la siguiente:

1. Identificación de cláusulas y operadores lógicos:

- Primero, se procesa la definición de la regla para identificar las cláusulas individuales que la componen.
- Además, se identifican los operadores lógicos (como AND y OR) que relacionan cada clausula con su sucesiva clausula.

2. Construcción del árbol:

- Cada cláusula individual se representa como un nodo hoja en el árbol de relaciones lógicas.
- Los operadores lógicos se representan como nodos internos del árbol, con los nodos hijos correspondientes a las cláusulas o subexpresiones lógicas que relacionan.

3. Manejo de anidamiento:

- En caso de que una cláusula contenga una expresión lógica anidada, esta se representa como un subárbol dentro del nodo correspondiente a la cláusula.
- De esta manera, se captura la estructura jerárquica de las expresiones lógicas, permitiendo manejar niveles de anidamiento arbitrarios.

4. Evaluación del árbol:

- Durante la evaluación de la regla, se recorre el árbol de relaciones lógicas, combinando los resultados de las cláusulas individuales según los operadores representados en los nodos internos.
- Este proceso permite obtener el resultado final de la regla, a partir de la evaluación de la estructura lógica completa.

La evaluación de una cláusula se determina de la siguiente forma:

ALGORITMO 2: Evaluación de una cláusula

```

1 FUNCIÓN evaluar_cláusula(cláusula, combinación_objetos):
2   izquierda, operador, derecha = cláusula
3   valor_izquierda = evaluar_valor(izquierda, combinación_objetos)
4   valor_derecha = evaluar_valor(derecha, combinación_objetos)
5   RETORNAR resultado de aplicar operador a valor_izquierda y valor_derecha

```

Donde la función evaluar valor retorna el atributo o ejecuta una función para obtener el atributo resultante.

Así podemos entender con un ejemplo si la regla contiene la siguiente expresión lógica:

((CLAUSULA1 AND CLAUSULA2) OR CLAUSULA3) (2)

Donde cada paréntesis refleja las secciones y orden que tendrían las relaciones lógicas. El árbol estaría representado por la **Figura 4** donde el árbol de relaciones lógicas tendría un nodo raíz de tipo 'OR' con dos nodos hijos: uno de tipo 'AND' que contiene CLAUSULA1 y CLAUSULA2, y otro nodo que contiene CLAUSULA3. Esto pues la primera sección sería (CLAUSULA1 AND CLAUSULA2) que se relaciona con un OR con la segunda sección (CLAUSULA3). Al evaluar esta estructura, primero se determinarían los resultados de CLAUSULA1, CLAUSULA2 y CLAUSULA3, luego se combinarían los resultados de CLAUSULA1 y CLAUSULA2 mediante el operador AND, y finalmente se evaluaría el operador OR entre este resultado y el de CLAUSULA3. En general el orden de recorrido del árbol responde al ordenamiento postorden (izquierdo, derecho, raíz).

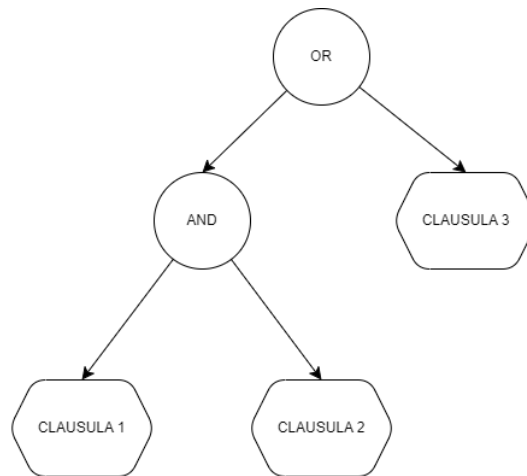


Figura 4: Ejemplo árbol relacional
Fuente: Elaboración propia

Esta arquitectura permite manejar expresiones lógicas anidadas y complejas, brindando flexibilidad y expresividad al lenguaje de dominio específico. Por otro lado, al separar la evaluación de las cláusulas individuales de la combinación de sus resultados según la lógica de la regla, se logra una mejor modularidad y mantenibilidad del código.

3.11 EVALUACIÓN DE REGLA:

El chequeo de propiedades, de relaciones y de composición lógica de las cláusulas y operadores respecto a múltiples instancias de objetos relevantes, determina el resultado de una instancia. Así los pasos para determinar el resultado dependerán de los resultados de cada instancia en las comprobaciones individuales de los objetos y las cláusulas, agregadas en función de las composiciones lógicas de la regla.

La evaluación de una regla implica un proceso estructurado que comienza con la identificación de los objetos relevantes, continúa con la evaluación de cada cláusula individual para cada combinación de objetos, y da cúlmene con la combinación de estos resultados parciales según la lógica representada en el árbol de relaciones.

3.11.1 IDENTIFICACIÓN DE OBJETOS RELEVANTES

Un aspecto crucial en el proceso de evaluación de reglas es la identificación de los objetos relevantes que participarán en dicha evaluación. La estructura de las reglas hace referencia a al menos una lista de objetos, los cuales se indica la relación entre los objetos según la lógica de los siguientes operadores:

- **TODO:** Este operador indica que la combinatoria de todos los objetos de cada lista asociado a él deben cumplir con las condiciones de la regla para que el resultado sea verdadero.
- **CUALQUIER:** Este operador indica que basta con que, en la combinatoria de cada objeto de cada lista asociadas a él, al menos una combinación entre objetos cumpla con las condiciones de la regla para que el resultado sea verdadero.

- **NINGUN:** Este operador indica que ningún objeto de cada lista asociada a él deben cumplir con las condiciones de la regla para que el resultado sea verdadero.

A través del proceso de evaluación, se generan todas las combinaciones posibles de los objetos indicados por los operadores, creando una matriz con tantas filas como la multiplicación entre las cantidades de objetos contenidos en cada lista de objetos. Y tendrá tantas columnas como objetos descritos.

	ALGORITMO 2: Ejemplo regla combinatoria
1	REGLA:
2	TODO lista1, lista2
3	CUALQUIER lista3
4	EN
5	CLAUSULA1 AND
6	CLAUSULA2 OR
7	CLAUSULA3
8	TERMINAR

Supongamos que:

- lista1 tiene 2 objetos: [obj11, obj12]
- lista2 tiene 2 objetos: [obj21, obj22]
- lista3 tiene 2 objetos: [obj31, obj32]

La matriz de objetos relevantes tendría las siguientes dimensiones:

- Filas: 2 (cantidad de objetos en lista1) x 2 (cantidad de objetos en lista2) x 2 (cantidad de objetos en lista3) = 8 filas
- Columnas: 3 (una columna por cada lista)

La matriz resultante se representa en **Tabla 23**

lista1	lista2	lista3
obj11	obj21	obj31
obj11	obj21	obj32
obj11	obj22	obj31
obj11	obj22	obj32
obj12	obj21	obj31
obj12	obj21	obj32
obj12	obj22	obj31
obj12	obj22	obj32

Tabla 23: Ejemplo matriz combinatoria objetos

Fuente: Elaboración propia

Esta matriz de objetos relevantes será utilizada posteriormente durante la evaluación de las cláusulas de la regla donde todas las repuestas a las relaciones de las cláusulas entre obj11, obj12, obj21, obj22 deben ser verdaderas y cualquier respuesta entre los objetos (obj11, obj21), (obj11, obj22), (obj12, obj21), (obj12, obj22) y los obj31, obj32 deben ser verdaderas.

3.11.2 EVALUACIÓN DE LA INSTANCIA MATRIZ

Una vez identificados los objetos relevantes y generado la matriz de todas las posibles combinaciones de los objetos relevantes, se agrega como columnas referencias a cada cláusula de la regla de modo que luego de las primeras columnas que representan objetos, las siguientes corresponde a las cláusulas y finalmente se agrega una última columna resultado, que representa el valor booleano de la evaluación de todas las cláusulas respecto a la composición lógica de la regla.

Para cada fila de la de evaluación, se evalúa con los objetos de la fila cada cláusula retornando el valor booleano almacenándose en la correspondiente celda de la matriz. Donde la evaluación de la cláusula corresponde a comparar los atributos o propiedades de los objetos de la de la fila correspondiente utilizando el operador lógico definido en la cláusula. Paralelamente, se construye el árbol de relaciones lógicas que representará la estructura de las operaciones lógicas entre clausulas.

Una vez se ha evaluado la matriz para cada combinación de objetos, se procede a recorrer el árbol de relaciones lógicas, para cada fila se combinan los resultados de las cláusulas según la operación lógica representadas en el árbol. El resultado de cada fila se almacena en la última columna de la matriz, que representa el resultado booleano de la aplicación de la regla para esa combinación de objetos.

El resultado final de la regla se obtiene de la relación de objetos especificada (TODO, CUALQUIER o NINGUNO) y el resultado de cada fila, donde si existe la relación “TODO”, el resultado será verdadero si todas las filas de los objetos con esta relación de la matriz, en cualquier otro caso será falso. Si existe la relación “CUALQUIERA”, el resultado será verdadero si al menos una de las filas de los objetos relacionados es verdadero, si son todos falsos, entonces el resultado será falso. Si la relación es “NINGUNO”, el resultado será verdadero si todas las filas de los objetos relacionados tienen un resultado falso, cualquier otro caso es falso el resultado de la matriz.

ALGORITMO 3: Evaluación regla

```

1 FUNCIÓN evaluar_regla(regla):
2   CREAR matriz_evaluación con filas = 0 y columnas = número de cláusulas + 1
3   CREAR árbol_relaciones_lógicas
4   cláusulas = obtener_clausulas(regla)
5
6   matriz_combinaciones = generar_combinaciones_objetos(regla)
7
8   PARA CADA combinación EN matriz_combinaciones:
9     AGREGAR combinación COMO nueva fila en matriz_evaluación
10
11   PARA CADA cláusula EN cláusulas:
12     resultado_cláusula = evaluar_cláusula(cláusula, combinación)
13     AGREGAR resultado_cláusula EN última columna de la fila actual
14
15   AGREGAR resultado de cada cláusula AL árbol_relaciones_lógicas
16   resultado_fila = evaluar_árbol(árbol_relaciones_lógicas, fila)
17   AGREGAR resultado_fila EN última columna de la fila
18
19   relación_objetos = obtener_relación_objetos(regla)
20   resultado_final = evaluar_resultado_final(matriz_evaluación, relación_objetos)
21
22   RETORNAR resultado_final

```

3.12 ASIGNACIÓN, EJECUCIÓN Y CONSTRUCCIÓN DE OBJETOS REALES Y VIRTUALES:

Entiéndase como un objeto real aquellos objetos directamente referenciados en el archivo IFC, es decir, todos los elementos con representación tridimensional del proyecto. Por lo que un objeto virtual son elementos virtuales creados e instanciados por la necesidad de la ejecución de la regla. Un ejemplo sería los planos rasantes necesarios para evaluar el espacio disponible para construir en un terreno, ya que estos objetos no son necesariamente referenciados con una representación tridimensional y es requerido para la evaluación, por lo que entonces es necesario crearlos según las especificaciones regionales dadas por el OGUC 2.6.3 [MINVU, 2016].

Los objetos reales se extraen del archivo IFC y son almacenados en un diccionario, donde las claves corresponden a los tipos IFC de los elementos (por ejemplo, "*IfcWall*", "*IfcWindow*", etc.) y los valores son listas que contienen la información geométrica de los objetos.

Para poder referenciar objetos en las reglas definidas en el DSL, el sistema debe proporcionar mecanismos que permitan acceder, manipular y almacenar estas estructuras de datos.

3.12.1 MANEJO DE VARIABLES:

El sistema cuenta con un mecanismo de asignación de variables que permite almacenar y utilizar valores durante la ejecución de las instrucciones. Estas variables pueden estar asociadas a objetos 3D, resultados de cálculos o cualquier otro tipo de dato necesario para la evaluación de la regla. La asignación de variables se realiza utilizando un nombre de variable seguido por el operador "=" que internamente ejecuta la instrucción "ASIGNAR".

Cuando necesitamos hacer referencia a una o más listas de objetos, el sistema es capaz de extraer las listas de objetos del diccionario de objetos geométricos del IFC. Esto se realiza utilizando la función "ELEGIR", la cual permite seleccionar y recuperar los objetos correspondientes a una determinada clase IFC a través del nombre de la clase deseada (por ejemplo, "IfcWall" o "IfcWindow").

Una vez obtenidas las listas de objetos relevantes, el sistema ofrece funciones que permiten realizar operaciones sobre estas listas. Por ejemplo, la función "SUMA" permite combinar varias listas de objetos en una sola lista unificada, mientras que la función "RESTA_LINEA" permite eliminar segmentos de líneas que se superponen entre dos listas de objetos lineales. Las funciones creadas en este prototipo se encuentran en **Tabla 24**.

Nombre	Descripción
ELEGIR	Permite seleccionar y recuperar los objetos correspondientes a una determinada clase IFC (por ejemplo, "IfcWall" o "IfcWindow") desde el diccionario de objetos 3D.
LINEA_CARA	Genera una lista de objetos 3D lineales que representan los bordes de una cara específica (superior o inferior) de un objeto 3D.
RESTA_LINEA	Elimina los segmentos de líneas que se superponen entre dos listas de objetos lineales, devolviendo una nueva lista con los segmentos restantes.
MAX_ALTURA	Calcula la altura máxima de un objeto 3D, ya sea lineal o triangulado, respecto al plano XY con $z = 0$.
PLANO	Construye un plano 3D a partir de una lista de objetos 3D lineales, con un ángulo y altura específicos.
LINEA_ES_SOBREPUESTO	Determina si una lista de objetos 3D lineales se superpone con otra lista de objetos 3D lineales, devolviendo una nueva lista con los segmentos superpuestos.
SUMA	Combina varias listas de objetos 3D en una sola lista unificada.

Tabla 24: Funciones de manipulación de objetos

Fuente: Elaboración propia

3.12.2 EJECUCIÓN DE INSTRUCCIONES

Durante el proceso de evaluación de una regla, el sistema ejecuta las instrucciones que componen dicha regla. Estas instrucciones se extraen a partir del texto de la regla, donde se define la sintaxis y estructura de cada instrucción. El sistema procesa el texto de la regla, identificando y separando cada instrucción individual, ya sea una asignación de variables, una evaluación condicional, una llamada a función o estructuras, tales como "REGLA" o "CONSTRUIR".

Las instrucciones extraídas se almacenan en una estructura de pila. Cada pila puede contener tanto instrucciones individuales como sub-pilas anidadas. Este suceso permite manejar la estructura jerárquica y la dependencia entre las diferentes instrucciones que componen la regla según las tabulaciones. Cuando una instrucción requiere la ejecución de un conjunto de sub-instrucciones, tal como "REGLA" o "CONSTRUIR", estas se almacenan en una pila que se apila dentro de la pila principal, creando así una estructura de pilas anidadas.

Esta organización de las instrucciones en pilas permite al sistema mantener el orden y la lógica de ejecución durante la evaluación de la regla. Además, la anidación de pilas facilita el manejo de secciones más complejas de la regla, como la definición de una "REGLA" que a su vez contiene múltiples cláusulas y operadores lógicos.

La instrucción "CONSTRUIR" permite crear objetos virtuales a partir de las listas de objetos existentes y las funciones de manipulación de objetos. Los nuevos objetos pueden ser, por ejemplo, planos o elementos lineales que representen conexiones o intersecciones entre los objetos originales. Estas construcciones se retornan en la función y asignan al atributo o nombre dado en la línea de la instrucción "CONSTRUIR".

Dentro de la instrucción "CONSTRUIR", se pueden utilizar las expresiones condicionales "SI" y "SINO" para permitir la construcción de objetos en casos excepcionales. Esto brinda flexibilidad y la posibilidad de manejar situaciones específicas que puedan surgir durante la ejecución de las reglas. Un ejemplo de ello es la construcción de la línea de deslindes, pues existe la excepción que cuando existe un muro sobre dicha línea, entonces se considerará el espacio compartido entre ambos objetos se considera como un eje de adosamiento el cual obedece otras reglas respecto a los ejes de deslindes para los distanciamientos.

En resumen, la gestión de objetos 3D en este lenguaje de dominio específico se basa en un diccionario que almacena los elementos del modelo IFC, junto con un sistema de manejo de variables que permite almacenar y utilizar datos durante la ejecución de las reglas. El sistema proporciona funciones para acceder a estas estructuras de datos, manipular las listas de objetos y construir nuevos elementos geométricos, incluyendo la posibilidad de utilizar expresiones condicionales para manejar casos excepcionales. Esto permite que las reglas definidas puedan operar sobre la información 3D de manera flexible y eficiente, facilitando la validación y el análisis de los modelos de construcción.

3.13 EDITOR DE REGLAS:

La definición de las reglas se realiza a través de un archivo de texto que sigue una sintaxis específica. El sistema tiene la capacidad de detectar errores básicos sobre cómo se deben definir cada posible instrucción retornando mensajes de error previos a cualquier ejecución.

La estructura del archivo de "Reglas" se basa en una sintaxis simple y coherente, que permite a los usuarios crear y modificar las reglas de una manera intuitiva y flexible. Donde las reglas se definen dentro de una sección delimitada por las palabras clave "REGLA:" y "TERMINAR", lo que facilita la organización y la identificación de las diferentes reglas. Los objetos virtuales se definen con la palabra clave "CONSTRUIR nombre_objeto" y "RETORNAR" donde nombre_objeto será el nombre de la variable retornada por la función.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

El prototipo ha sido validado a través de una prueba de concepto desarrollado con un arquitecto con un perfil profesional con vasta experiencia relacionada a la obtención de permisos en DOM de todo el país, peritajes y asesoramientos legal a diversas empresas inmobiliarias. Adicionalmente a esto, considerando que el prototipo es un conjunto de funcionalidades y herramientas acorde al dominio de revisión para aprobación respecto a normas de construcción de proyectos arquitectónicos, se le puede considerar un motor de revisión y por ende se pueda validar respecto a cualquier proyecto si este responde identificando si cumple o no la norma con el cual se ejecute, por lo tanto, se utilizarán ejemplos de uso de modo que permita dar un mejor entendimiento a su utilidad actual.

Precisamente este proyecto se hará cargo de la aplicación de algunas normas urbanísticas, por lo que es necesario saber:

- **Línea Oficial:** Es el límite que separa el espacio público del privado, determinado en el plano regulador respectivo. Este límite puede estar constituido por la línea que marca la propiedad de una vía pública, un parque u otro espacio de uso público.
- **Línea de edificación:** es la línea que indica el límite hasta el cual se permite la construcción de edificios en un predio. Esta línea puede coincidir con la Línea Oficial o estar a una cierta distancia de ella, de acuerdo con las disposiciones del plano regulador o las normativas específicas aplicables a la zona.
- **Eje público:** se refiere al eje central de una vía pública, que es utilizado como referencia para la medición y definición de distancias y alineamientos en los planos reguladores.
- **Rasantes:** Las Rasantes son líneas que definen la pendiente máxima permitida para la edificación en relación con el terreno natural o la vía pública adyacente. Estas líneas determinan los niveles máximos de construcción para asegurar la correcta iluminación y ventilación de los edificios y evitar la obstrucción de vistas.
- **Eje de deslindes:** es la línea que marca la separación entre dos propiedades colindantes. Este eje es utilizado para determinar los límites de construcción y asegurarse de que se respeten las distancias mínimas establecidas entre edificaciones adyacentes, conforme a las normativas vigentes.
- **Adosamiento:** es la construcción de una edificación en contacto directo con la línea de deslinde de un predio, es decir, sin dejar separación o distancia alguna entre la construcción y el límite de la propiedad. Este tipo de construcción está regulado por normativas específicas que buscan asegurar la convivencia armónica entre edificaciones contiguas.

Además de mencionar cuales serán las normas las cuales será evaluadas con tal de validar los requisitos de este proyecto:

Regla	Descripción
Altura máxima de edificación Art. 2.1.23 OGUC	Se describe en la OGUC como la altura máxima a la cual se puede construir en un terreno respecto a nivel natural del terreno.
Distanciamientos a deslindes Art. 2.6.3 OGUC	Se describe en el OGUC como la distancia mínima que debe existir entre los deslindes de la propiedad y los muros de la construcción, depende de la altura del piso y si los muros poseen vanos o no.
Rasantes de propiedad Art. 2.6.2 y 2.6.3 OGUC	Se describe en el OGUC como planos inclinados con respecto al plano horizontal que se levantan desde el nivel del terreno en todos y cada uno de los puntos que forman los deslindes y el eje de la calle que enfrenta el predio. Donde no se podrá construir sobre las rasantes.
Espacio de antejardín Art. 2.5.8 OGUC	Se describe en el OGUC como el área entre la línea oficial y la línea de edificación, regulada en el instrumento de planificación territorial.

Tabla 25: Descripciones Reglas a evaluar

Fuente: Elaboración propia

Ya que en este trabajo se desarrolló un programa con las capacidades descritas en los objetos específicos de:

1. Desarrollar una plataforma que permita ingresar planos digitalmente e interpretarlos.
2. Contrastar planos respecto a las exigencias descritas en el certificado de informes previos semejante a como si lo resolviese un revisor municipal o independiente.
3. Ejecutar evaluaciones automáticamente, entregando como respuesta un listado de observaciones del plano en un lenguaje que aporte al arquitecto proyectista la posibilidad de corregir su propuesta con conocimiento de causa.

Se resolverá la validación teniendo en cuenta dichos objetivos por cumplir.

4.1 VALIDACIÓN DE SOFTWARE:

En esta sección se verificará que el software funcione correctamente, proporcionando una descripción técnica y pruebas unitarias para los componentes principales del sistema, pruebas de integración para asegurar que el código funcione correctamente y pruebas de sistemas para evaluar el software en lo que significa su totalidad.

4.1.1 DESCRIPCIÓN TÉCNICA:

El software desarrollado en Python tiene como objetivo principal evaluar el cumplimiento de reglas y normativas en modelos BIM, representados en archivos IFC. El sistema utiliza técnicas de procesamiento geométrico y análisis espacial para verificar si los elementos arquitectónicos y estructurales del proyecto cumplen con los criterios establecidos en un conjunto de reglas definidas por el usuario.

El flujo de trabajo del software se divide en dos componentes principales: el procesamiento de las reglas y la evaluación del modelo BIM. En general, esto se divide en dos librerías, “Lectura.py” se encarga de leer y procesar las reglas en un lenguaje en el lenguaje de dominio específico descrito en el Capítulo 3, mientras la librería “imprimir.py” se encarga de extraer las geometrías del modelo BIM y agregar o modificar componentes del modelo con objetivo de compatibilizarlo con las estructuras de datos establecidos para evaluar las reglas, esto también incluye las funciones para generar visualizadores de los objetos geométricos.

En la librería “Lectura.py”, se define una sintaxis específica para escribir las reglas que su evaluación será aplicada posteriormente. Estas reglas se componen de instrucciones que establecen relaciones entre objetos, definen las cláusulas condicionales y especifica acciones a realizar. El software utiliza técnicas de análisis léxico y sintáctico para procesar las reglas y generar una estructura de datos que representa la lógica de evaluación tal como fue descrito en el capítulo anterior.

Por otro lado, en “imprimir.py” se encarga de interactuar con la biblioteca “*ifcopenshell*” para leer y extraer la información geométrica del modelo BIM almacenado en un archivo IFC. Este componente utiliza algoritmos de procesamiento geométricos para calcular propiedades especiales y relevantes, como alturas, distancias y relaciones entre elementos arquitectónicos.

Las reglas, escritas en un DSL deben ser procesadas en términos computacionales de modo que se creen pilas con las instrucciones necesarias para computarlas, una vez se han procesado las reglas y extraído las geometrías del modelo BIM, el software combina ambas partes para realizar las evaluaciones de cumplimiento. Utiliza estructuras de datos generadas a partir de las reglas y aplica los cálculos geométricos necesarios sobre los elementos del modelo. El resultado de esta evaluación indica si el modelo cumple o no con las reglas establecidas.

Las pruebas que el software puede evaluar actualmente incluyen:

1. **Altura Máxima:** verificar si la altura de los elementos del modelo no excede un valor máximo permitido.
2. **Rasantes:** Comprueba si los elementos del modelo respetan los límites de altura en relación con planos inclinados definidos en las normativas OGUC.
3. **Antejardín:** Verificar si los elementos del modelo mantienen una distancia mínima respecto a los límites del terreno.
4. **Distanciamiento:** Evaluar si los elementos del modelo cumplen con las distancias mínimas requeridas entre sí. Excluyendo casos límites donde los vanos se encuentren a más de 1.8 [m] de altura respecto al piso del nivel.

En general, el software desarrollado es capaz de proporcionar herramientas poderosas para evaluar el cumplimiento de reglas y normativas en modelos BIM atinentes a las metodologías actuales, utilizadas por revisores independientes y municipales. Se usan técnicas de procesamiento de lenguaje y geometría computacional para analizar la información contenida en archivos IFC y generar resultados que ayuden a los profesionales a tomar decisiones informadas ya garantizar que los diseños cumplan con los estándares de regularizaciones aplicables.

4.1.2 DECLARACIONES DE NORMAS:

Se han creado 4 documentos donde cada uno representa una norma específica, donde la primera a explicar serán los más sencillas, para luego seguir con las más complejas.

1. **Altura máxima:** En general se identifica que los elementos requeridos en la altura máxima se relacionan simplificadaamente, entre las composiciones de los elementos estructurales de una casa (Muros y techo) y el terreno, ya que la metodología general para realizar dicho análisis es determinar la altura que existe entre cada elemento estructural respecto al nivel natural del terreno. Asumiendo que en el proyecto original el terreno debe estar ajustado a dichos niveles, entonces se debe calcular la altura de los elementos de CASA (TECHO y MURO) versus el terreno, esto se permite gracias a la función ALTURA_DESDE(Objeto desde =, objeto medido desde =, usar cara Superior objeto 1 =, usar cara superior del objeto 2 =). En general esta función nos dirá a qué altura máxima se encuentra cada elemento de casa desde la cara superior del terreno hasta la parte superior de cada muro o tejado. Luego si esta altura es inferior a 7 metros, entonces los elementos identificados cumplen la norma, en caso contrario incumplen la norma.

```
Reglas > ALTURA MAXIMA.txt
1 #ALTURA MAXIMA
2
3 TECHO = ELEGIR(IFCROOF)
4 MURO = ELEGIR(IFCWALL)
5 TERRENO = ELEGIR(IFCGEOGRAPHICELEMENT)
6 CASA = SUMA(TECHO, MURO)
7
8 REGLA:
9   TODO CASA, TERRENO
10  EN
11    ALTURA_DESDE(TERRENO, MURO, SUPERIOR, CUALQUIERA) <= 7
12    TERMINAR
13
```

Figura 5: Regla Altura Máxima

Fuente: Elaboración Propia

Un ejemplo de incumplimiento de esta norma tenemos el siguiente ejemplo:

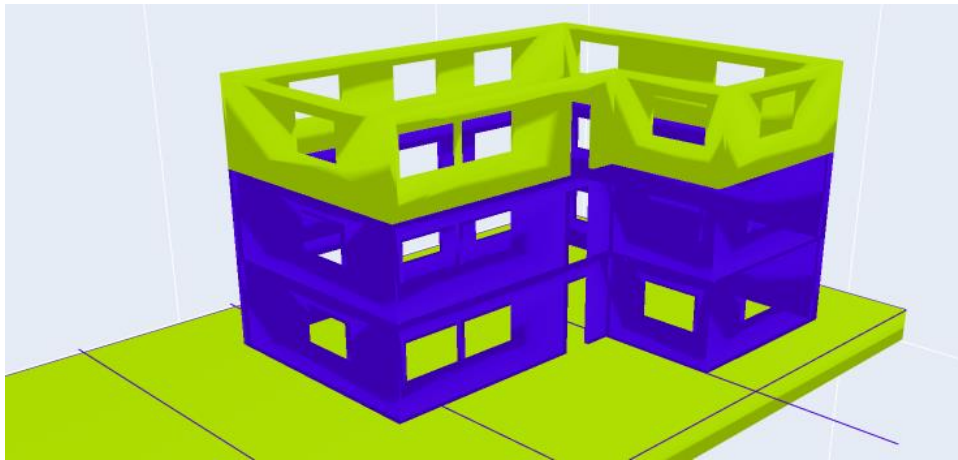


Figura 6: Ejemplo Altura máxima Incumple

Fuente: Elaboración propia

Donde se visualiza que los niveles pisos colorados amarillo están incumplimiento de la regla, pues se encuentran a una altura de 7.95 [m].

Podemos validar lo mismo revisando el archivo Revit del cual se obtuvo el proyecto:

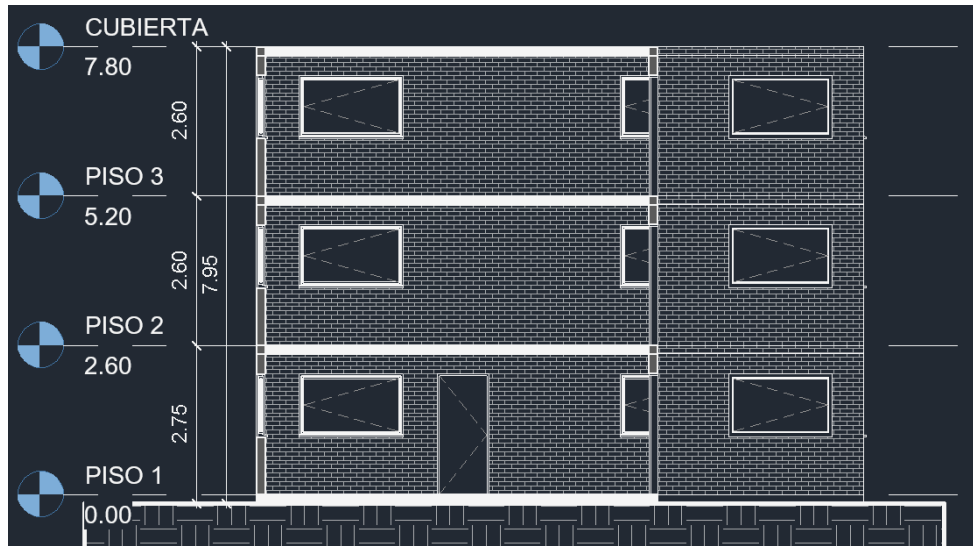


Figura 7: Proyecto Revit Max altura no cumple

Donde se nota que efectivamente la altura del tercer piso respecto al piso es efectivamente de 7.95 [m] y por ende incumple de tener una altura inferior a los 7.0 [m].

Además, para el caso donde cumple la condición de que la altura máxima cumple lo descrito por la norma tenemos:

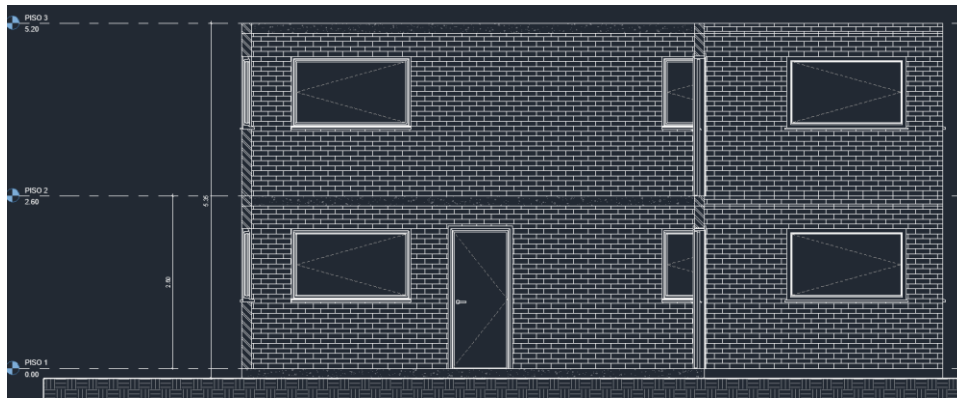


Figura 8 Proyecto Revit Max altura Cumple

Su altura máxima de edificación es de 5.35 [m] por lo que si cumple la condición de la regla y donde la salida del prototipo es:

Resultado de la regla: True

Por lo que no revelará ninguno de los visualizadores ya que no existen elementos fallando la norma que mostrar.

- 2. Antejardín:** En esta regla se deben considerar los objetos estructurales básicos del proyecto, TECHO y MUROS, resumidos como elementos de CASA. Adicionalmente, se requiere precisar que este proyecto se hará cargo de las aplicaciones de algunas dichas de la norma urbanística, por lo que se ha de saber lo que es línea oficial y línea de edificación, siendo la primera la línea que describe el límite entre la propiedad privada del terreno donde se planea edificar y el espacio público (calle) y la segunda se refiere a la línea desde donde está permitida la construcción, el espacio formado entre la línea oficial y de edificación denota el espacio de antejardín. Donde se debe considerar que no se permite ningún tipo de edificación en el espacio de antejardín, por lo que considerando el estado de no poder atravesar la línea de edificación se crea un eje con ella como base con una inclinación de 90 grados respecto a la horizontal natural utilizando la función PLANO(Linea_base =, angulo_inclinacion =, altura desde donde inicia inclinación =, Altura máxima =). Finalmente, la regla descrita es la siguiente:

```
Reglas > ANTEJARDIN.txt
1  #ANTEJARDIN
2
3  L_EDIFICACION = ELEGIR(IFCANNOTATION['LINEA DE EDIFICACION'])
4  L_OFICIAL = ELEGIR(IFCANNOTATION['LINEA OFICIAL'])
5  MUROS=ELEGIR(IFCWALL)
6  MAX_ALTURA_MURO = MAX_ALTURA(MUROS)
7  PLANO_RASANTE_EDIFICACION=PLANO(L_EDIFICACION, 90, 0, MAX_ALTURA_MURO)
8
9  REGLA:
10  |  TODO L_EDIFICACION, L_OFICIAL, PLANO_RASANTE_EDIFICACION, MUROS
11  |  EN
12  |  DISTANCIA_LINEA(L_EDIFICACION, L_OFICIAL) >= 5 AND
13  |  PLANO_ES_SOBREPUESTO(PLANO_RASANTE_EDIFICACION, MUROS) == FALSE
14  |  TERMINAR
```

Figura 9: Regla Antejardín
Fuente: Elaboración Propia

Donde se puede ver que se deben crear planos que representen las rasantes sobre la línea de edificación, las líneas de edificación, línea oficial y muros.

Donde la primera cláusula determina que si la distancia entre las líneas de edificación y oficial tiene una distancia mayor a 5 [m] y la segunda cláusula determina que, si existe un elemento de muros que se sobreponga o intercepte al plano de la línea de intersección, pues en caso de atravesarlo significa que los muros ocupan espacio del antejardín.

Un ejemplo de incumplimiento es:

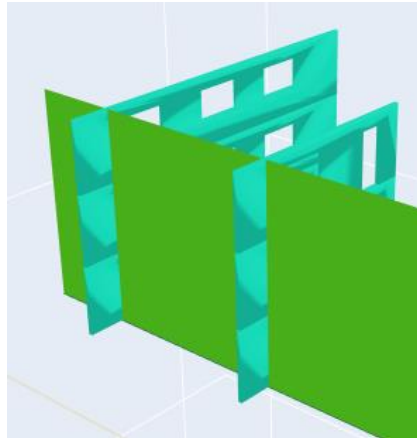


Figura 10: Ejemplo antejardín incumple

Fuente: Elaboración propia

Donde notamos en verde el eje de la línea de edificación y los muros celeste atraviesan dichos muros incumpliendo el espacio de no construcción del antejardín. Esto es verificable observando el proyecto en Revit original.

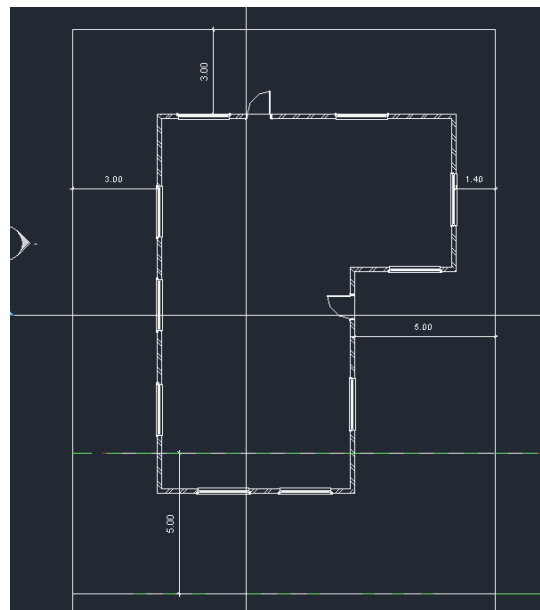


Figura 11: Proyecto Antejardín No cumple

Fuente: Elaboración Propia

Donde en el plano de planta de la figura **Figura 11** es observable que la distancia entre las líneas oficiales y de edificación cumplen el distanciamiento de 8 metros más el visualizador solo muestra los muros que incumpliendo, atravesando el eje de edificación.

Además, para el caso donde cumple la condición de que del antejardín cumple lo descrito por la norma tenemos:

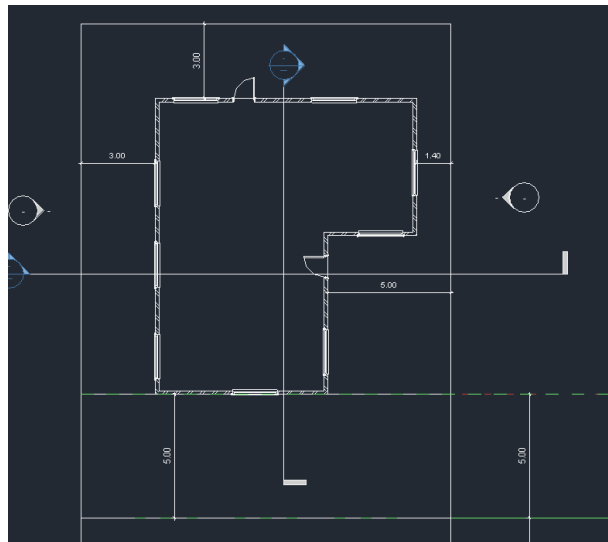


Figura 12: Proyecto Antejardín Cumple
Fuente: Elaboración propia

Donde es observable la distancia entre líneas es igual a 5 [m] y ningún muro atraviesa la línea de edificación como el caso anterior, por lo que la salida de nuestro sistema es.

Resultado de la regla: True

Lo cual es correcto dado el proyecto.

3. **Rasantes:** Para la creación de las rasantes, hay que tener en consideración que las rasantes están condicionadas a la existencia de viviendas aislada o adosadas, siendo estas donde existe un muro con ciertas características de altura y resistencia ígnea que divide el terreno propio del de un vecino, donde dependiente del tipo de deslinde existente (deslinde del terreno o adosado), existen distintas condiciones para la formación de rasantes, donde para la quinta región y rasantes de deslindes al terreno deben estar ajustadas a las líneas de deslindes y tener una inclinación de 70 grados para la quinta región y en caso de estar en un deslinde adosado, debe considerarse un plano de 3.5 metros de altura con 90 grados desde donde empieza un segundo plano con una inclinación de 45 grados. Para poder construir esto es necesario detectar donde existen muros de adosamientos, si es que existen, para ello se aplican funciones condicionales SI o SINO, donde si existe una Línea de adosamientos, entonces construye las rasantes teniendo en consideración los ejes de adosamientos y SINO, entonces construye las rasantes sobre los deslindes del terreno asumiendo que es una casa aislada.

```

Reglas > RASANTES.bt
1 #RASANTES
2
3 MUROS=ELEGIR(IFCWall)
4 TECHO=ELEGIR(IFCRoof)
5 CASA = SUMA(MUROS, TECHO)
6
7 CONSTRUIR DOMO:
8
9 L_BORDE_TERRENO=ELEGIR(GEOGRAPHICALUPPERBOUND)
10 L_EDIFICACION=ELEGIR(IFCANNOTATION['LINEA DE EDIFICACION'])
11 L_EJEPUBL=ELEGIR(IFCANNOTATION['EJE PUBLICO'])
12 L_INF_MURO=LINEA_CARA(MUROS, INFERIOR) #LINEA_CARA(OBJ_3D =, CARA =)
13 L_DESLINDES=RESTA_LINEA(L_BORDE_TERRENO, L_INF_MURO, 0.0001) #RESTA_LINEA(LINEA1 =, LINEA2 =, DELTA =)
14 MAX_ALTURA_MURO = MAX_ALTURA(CASA)
15
16 PLANO_RASANTE_EDIFICACION=PLANO(L_EDIFICACION, 90, 0, MAX_ALTURA_MURO) #PLANO(USAR_BASE =, ANGULO =, DESDE_ALTURA =, MAX =)
17 PLANO_RASANTE_EJEPUBLIC=PLANO(L_EJEPUBL, 70, 0, MAX_ALTURA_MURO) #PLANO(USAR_BASE =, ANGULO =, DESDE_ALTURA =, MAX =)
18
19 L_ADOSADA = LINEA_ES_SOBREPUESTO(L_BORDE_TERRENO, L_INF_MURO, 0.0001) #LINEA_ES_SOBREPUESTO(OBJ_PLANO1 =, OBJ_PLANO2 =, DELTA =)
20
21 SI L_ADOSADA:
22 PLANO_RASANTE_DESLINDES=PLANO(L_DESLINDES,70, 0, MAX_ALTURA_MURO) #PLANO(USAR_BASE =, ANGULO =, DESDE_ALTURA =, MAX =)
23 PLANO_RASANTE_ADOSADOS=PLANO(L_ADOSADA, 45, 3.5, MAX_ALTURA_MURO) #PUEDE SER 0 SI USAR_BASE = 0
24
25 DOMO = SUMA(PLANO_RASANTE_EDIFICACION, PLANO_RASANTE_EJEPUBLIC, PLANO_RASANTE_DESLINDES,PLANO_RASANTE_ADOSADOS)
26 RETORNAR DOMO
27
28 SINO:
29 PLANO_RASANTE_DESLINDES=PLANO(L_DESLINDES,70, 0, MAX_ALTURA_MURO) #PLANO(USAR_BASE =, ANGULO =, DESDE_ALTURA =, MAX =)
30 DOMO = SUMA(PLANO_RASANTE_EDIFICACION, PLANO_RASANTE_EJEPUBLIC, PLANO_RASANTE_DESLINDES)
31 RETORNAR DOMO
32
33
34 REGLA:
35 TODO DOMO, MUROS
36 EN
37 PLANO_ES_SOBREPUESTO(DOMO, MUROS) == FALSE
38 TERMINAR
39

```

Figura 13: Regla de Rasantes

Fuente: Elaboración propia.

En términos generales la regla para la construcción sigue la lógica de creación de cualquier revisor, a pesar de que estos tienen la capacidad de identificar situaciones similares con su percepción y entrenamiento, este código no permite espacio de interpretación, ya que se ajusta a las instrucciones previas para crear las rasantes.

Ahora la regla consiste en que, si cualquier elemento del proyecto CASA atraviesa o se superpone a cualquiera de las rasantes, entonces incumple la norma. Un ejemplo de incumplimiento es el siguiente.

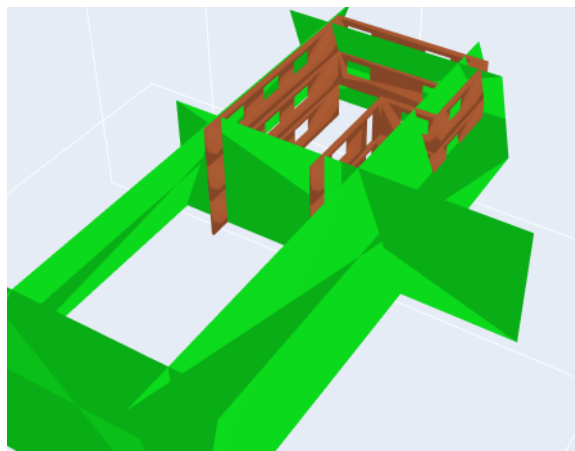


Figura 14: Ejemplo incumplimiento de rasantes

Fuente: Elaboración propia

Donde las rasantes se representan en color verde y los muros que incumplen son aquellos naranjas, vemos que en su mayoría los muros incumplen el no atravesar algunas rasantes. Verificando respecto al proyecto Revit, tenemos:

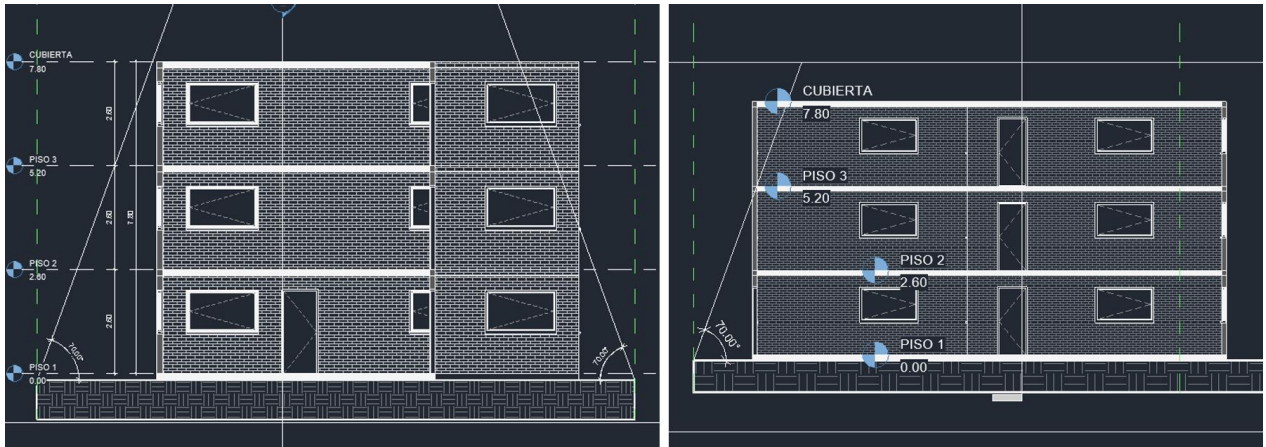


Figura 15: Proyecto rasantes no cumplen
Fuente: Elaboración Propia

Donde podemos ver que 2 rasantes incumplen la norma, sin incluir que atraviesa la línea de edificación, construyendo en el antejardín. Por lo que es congruente con la salida de nuestro prototipo.

Además de lo anterior, es un caso de estudio aparte la consideración en el caso de existencia de adosaminetos. Por lo que en el siguiente proyecto con adosamiento.

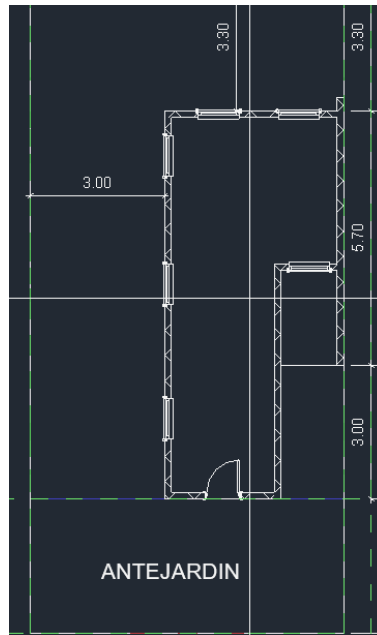


Figura 16: Proyecto que incumple rasantes con adosamiento
Fuente: Elaboración Propia

Donde el resultado de nuestro prototipo indica que el tejado y el muro adosado incumple, lo cual es correcto debido a que dicho muro supera los 3.5 [m] de altura. Y El tejado atraviesa una rasante de 70 grados desde los deslindes.

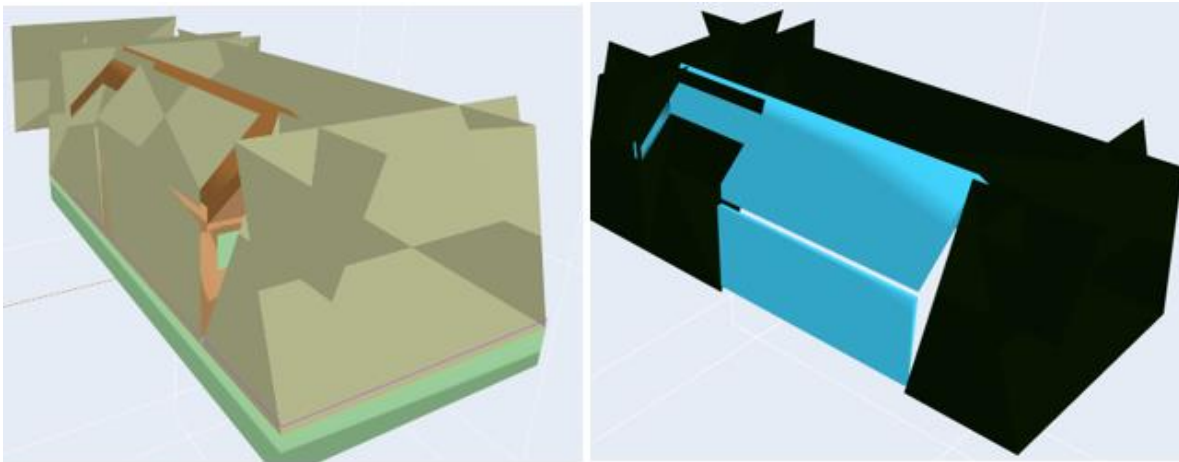


Figura 17: Ejemplo incumplimiento de Rasante con adosamiento

Fuente: Elaboración propia

Finalmente para ambos casos en los proyectos donde cumple la norma, recibimos la salida de **regla: True**, lo cual se condice correctamente con lo esperado.

Resultado de la regla: True

4. Distanciamientos:

Para el cálculo de distanciamientos se debe considerar que existentes diferentes tramos que se describen previamente en la **Tabla 14**, tal como se describe en la sección de **3.7 Sintaxis de reglas:**, se deben crear los distintos tramos respecto a las alturas versus si el elemento posee vano o no. Para poder lograr ello, se deben utilizar elementos como Muros, Planos_deslindes y terreno para poder calcular las alturas relativas.

```

Reglas > DISTANCIAMIENTO.txt
1  #DISTANCIAMIENTO
2
3  MUROS = ELEGIR(IFCWALL)
4  L_BORDE_TERRENO=ELEGIR(GEOGRAPHICALUPPERBOUND)
5  L_INF_MURO=LINEA_CARA(MUROS, INFERIOR)
6  L_DESLINDES=RESTA LINEA(L_BORDE_TERRENO, L_INF_MURO, 0.0001)
7  MAX_ALTURA=MAX_ALTURA(MUROS)
8  TERRENO = ELEGIR(IFCGEOGRAPHICELEMENT)
9
10 PLANO_DESLINDES = PLANO(L_DESLINDES, 90, 0, MAX_ALTURA)
11
12 REGLA:
13     TODO MUROS, PLANO_DESLINDES
14     CUALQUIER TERRENO
15     EN
16     ALTURA_DESDE(TERRENO, MUROS, SUPERIOR, SUPERIOR) <= 3.5 AND
17     FACHADA_CON_VANO(MUROS) == TRUE AND
18     DISTANCIA_PLANO(MUROS, PLANO_DESLINDES) >= 3
19     OR
20     ALTURA_DESDE(TERRENO, MUROS, SUPERIOR, SUPERIOR) <= 3.5 AND
21     FACHADA_CON_VANO(MUROS) == FALSE AND
22     DISTANCIA_PLANO(MUROS, PLANO_DESLINDES) >= 1.4
23     OR
24     ALTURA_DESDE(TERRENO, MUROS, SUPERIOR, SUPERIOR) >= 3.5 AND
25     ALTURA_DESDE(TERRENO, MUROS, SUPERIOR, SUPERIOR) <= 7 AND
26     FACHADA_CON_VANO(MUROS) == TRUE AND
27     DISTANCIA_PLANO(MUROS, PLANO_DESLINDES) >= 3
28     OR
29     ALTURA_DESDE(TERRENO, MUROS, SUPERIOR, SUPERIOR) >= 3.5 AND
30     ALTURA_DESDE(TERRENO, MUROS, SUPERIOR, SUPERIOR) <= 7 AND
31     FACHADA_CON_VANO(MUROS) == FALSE AND
32     DISTANCIA_PLANO(MUROS, PLANO_DESLINDES) >= 2.5
33     TERMINAR
    
```

Figura 18: Regla Distanciamiento

Fuente: Elaboración propia

En grosso modo, se construye los ejes de los deslindes respecto a sus definiciones y se crean cada sección de los descrito en **Tabla 14** entre cada OR de la regla. En general, se debe buscar las alturas de los muros respecto a los terrenos para identificar en qué tramos pertenecen, para luego determinar si los elementos tienen vano o no para finalmente detectar si los muros se encuentran a la distancia correspondiente a los planos deslindes.

Los resultados en las pruebas se ven:

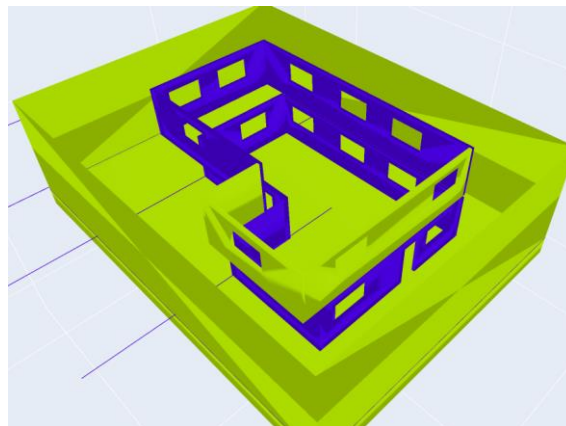


Figura 19: Ejemplo incumplimiento distanciamiento

Fuente: Elaboración Propia

En el cual todos los elementos amarillos se detectan como relacionados al fracaso de la norma de distanciamientos, donde para especificar, los muros que sobresalen del segundo piso de la propiedad incumplen las normas de distanciamiento. Ello se puede comprobar en el plano original.

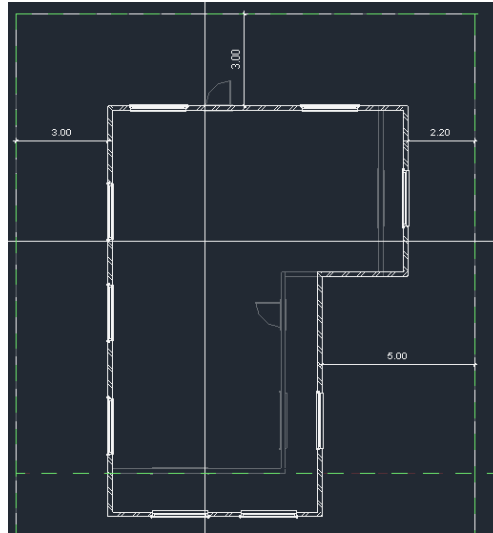


Figura 20: Proyecto que incumple Distanciamiento
Fuente: Elaboración propia

En la **Figura 20** se puede notar que existe un muro del segundo piso que no cumple el distanciamiento al estar a 2.20 [m] del deslinde, además de los muros extendidos para cerrar con dicho muro, donde la distancia a sus cantos también es de 2.20[m]. Tal como muestra el resultado del prototipo.

En el caso que el proyecto cumple el distanciamiento se recibe la respuesta:

Resultado de la regla: True

Finalmente, respecto a todos los ejemplos presentados anteriormente, se puede ver que el prototipo cumple correctamente la revisión de las normas establecidas de distanciamientos, rasantes, altura máxima y distanciamiento de antejardín. Así cumple las expectativas que el software funcione correctamente respecto a lo especificado en los objetivos generales.

4.2 VALIDACIÓN DE HIPÓTESIS:

Refiriéndose a la hipótesis dada en los objetivos específicos, es menester resumir que el prototipo creado cumple todos los puntos mencionados en dicha sección. En general, el prototipo interpreta cualquier proyecto arquitectónico a través de su modelado 3d y la clasificación de sus elementos gracias al estándar IFC, permitiendo digitalizar cualquier plano, pues el modelo obtenido consiste en una representación a escala de todo el proyecto, donde si se desea interpretar plan métricamente solo se deben crear las funciones necesarias para dicho objetivo. Se ha creado un lenguaje de dominio específico que permite virtualizar los pasos necesarios para comprobar cualquier norma computable.

Finalmente, cada ejecución está totalmente automatizada, donde el usuario solo requiere ingresar el proyecto IFC y el archivo que contiene las instrucciones para la regla analizada y luego de la ejecución el sistema entregará las salidas necesarias para visualizar e identificar los elementos que incumplen los indicado en la regla.

4.2.1 INTERPRETACIÓN DIGITAL DE PLANOS:

Con tal de demostrar la capacidad del software, el cual tiene que interpretar digitalmente un proyecto arquitectónico, hay que entender cómo se describe un archivo IFC y cómo se interpretan los elementos de ello. Como el sistema capta dicha interpretación, almacena, qué metodología de interpretación de los proyectos se utiliza y como se permite crear, identificar o modificar cualquier elemento con representación geométrica.

Es importante mencionar que un paradigma de interpretación de planos escritos en PDF o en papel, resulta ineficiente al momento de por computar e identificar sus elementos con precisión, pues para un computador un plano en dichos formatos tendría que ser procesados como imágenes, cuestión que puede crear errores de interpretación como la pérdida de información dimensional, ya que visualizar las distintas dimensiones se requiere el uso de múltiples planos que combinados entregan la información suficiente, otra desventaja es la falta de estructura semántica, ya que no existen diferencias entre líneas que permiten identificar con precisión a que clasificación pertenece cada elemento, otro inconveniente son las limitaciones visuales y manipulación, por lo que con ellos no se puede crear nuevos cortes o secciones que puedan resultar útiles para interpretar un proyecto en distintos ángulos. Así existen aún más dificultades inadecuadas a lo requerido para poder interpretar computacionalmente, por ende, en este trabajo se prefiere un paradigma que permite la interpretación tridimensional de cada proyecto, ya que de este modo se exporta con precisión las geometrías de los elementos y facilidad de diferenciarlos entre sí.

Para validar se utilizará un proyecto real diseñado en Reñaca, en la región de Valparaíso.



Figura 21: Proyecto Reñaca obtenido desde Revit 2025

Fuente: Diseño original Jorge Narea Toro [Narea, 2023]

Con tal de validar el proyecto, el sistema debe tener la capacidad de representar todos los elementos con representación tridimensional del proyecto y poder clasificarlos según el estándar IFC descritos en **Tabla 26**. Estos nos son todos los elementos con representación geométrica que posee el proyecto, mas son aquellos que han resultado útiles para la interpretación de reglas.

Clase IFC	Color Representación	Descripción
IfcGeographicElement	Rojo	Elementos que representan tridimensionalmente terreno
IfcSpace	Café	Elementos que representan Volúmenes de cada habitación de la propiedad
IfcWall	Turquesa	Elementos clasificados como Muros
IfcColumn	Morado Claro	Elementos clasificados como columnas
IfcSlab	Morado Oscuro	Elementos identificados como pisos
IfcDoor	Verde Lima	Elementos identificados como puertas
IfcWindow	Morado Ligeramente Claro	Elementos identificados como ventanas
IfcRoof	Violeta	Elementos identificados como techos
IfcAnnotation	Cada anotación tiene un color único.	Elementos de anotación, pueden ser líneas o planos, frecuentemente utilizados para generar referencias.

Tabla 26: Clases IFC por validar

Fuente: Elaboración propia

El primer paso para conseguir los datos necesarios en formato IFC, es exportar los archivos desde el programa de metodología BIM de preferencia. En Revit 2025 se utilizan las configuraciones de dividir muros, pilares, conductos por nivel. Utilizando la versión IFC Reference View y exportar elementos 2D de vista de plano. Ya con el archivo IFC exportado, simplemente se entrega la ubicación del archivo para que inicie el proceso de extracción y clasificación de los elementos.

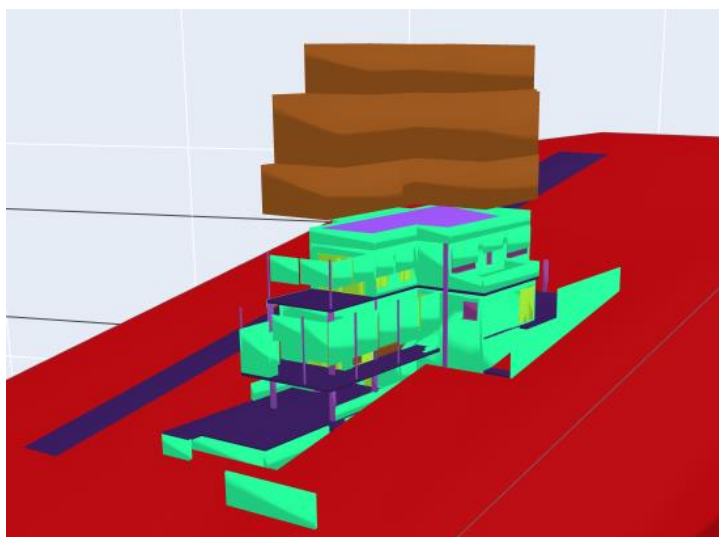


Figura 22: Proyecto digitalizado IFC

Fuente: Elaboración propia

Se puede notar que de la **Figura 22** se han obtenido todos los elementos posicionados correctamente, aficionados a volúmenes de los espacios interior, donde de dichos elementos aquellos que representan el volumen de todo un piso se posicionan encima de la casa, por otro lado, los volúmenes que representan cada habitación están posicionados ocupando el volumen de cada habitación dentro de la casa.

En caso de necesitar extraer más elementos se tiene control para elegir los elementos cargados al sistema en cada ejecución.

```
clases_ifc = ['IfcGeographicElement',  
             'IfcSpace',  
             '#IfcFlowTerminal',  
             '#IfcOpeningElement',  
             '#IfcFooting',  
             '#IfcCovering',  
             'IfcWindow',  
             '#IfcBeam',  
             '#IfcBuildingElementProxy',  
             'IfcRoof',  
             '#IfcCurtainWall',  
             '#IfcRailing',  
             'IfcSlab',  
             '#IfcFurniture',  
             '#IfcPlate',  
             'IfcDoor',  
             'IfcColumn',  
             '#IfcStairFlight',  
             'IfcAnnotation',  
             '#IfcMember',  
             'IfcWall']
```

Figura 23: Clases IFC obtenidas

Fuente: Elaboración propia

O para obtener todos los elementos que posee un proyecto se puede utilizar la función `obtener_clases_geometricas(archivo_ifc)` el cual retornará todas las clases IFC que posean una representación geométrica. Además, teniendo control de obtener cada elemento de cada clase separadamente, es posible obtener las propiedades de cada elemento, de modo que, si tenemos dos tipos de muros, uno de ladrillo y otro de metalcon, se pueden obtener sus por separado con tal de contar con las características especiales como grosos, nombre de tipo de muro, materiales, etc. en caso de ser necesario.

Así se puede decir que efectivamente el sistema creado tiene la capacidad de interpretar tridimensionalmente y las características que posee cada elemento del proyecto. Al ser valores geométricos descrito por vértices y aristas, es fácil poder obtener, modificar o crear cualquier elemento y describirlo. Por otra parte, los ejes coordenados están escalados 1:1 donde el eje x+ representa el norte, el eje x- el sur, el eje y+ representa el este y el eje y- el oeste. Finalmente, el plano xy con $z = 0$ corresponde al nivel natural del terreno.

4.2.2 CONTRASTE DE PROYECTO CON NORMAS

Con tal de validar la capacidad del software de contrastar planos respecto a las exigencias descritas en el certificado de informes previos, semejante a como si lo resolviese un revisor municipal o independiente, depende de las instrucciones creadas en el lenguaje de dominio específico, ya que el sistema ofrece la libertad de crear las instrucciones para cada regla con un DSL la validez de las normas depende de la capacidad del escritor de traducirlas en este lenguaje. Para este informe se ha mostrado y explicado en los puntos anteriores las reglas creadas para validar el sistema, se ha enseñado la sintaxis necesaria entender dicho lenguaje y en caso de ser necesario, crear más reglas, donde el único límite son las funciones específicas que se puedan requerir dado los comportamientos. Por ejemplo, una función que permita calcular el área de la base de un volumen, de modo que se puedan calcular el área de los *IfcSpace* tal que se puedan calcular coeficientes de constructibilidad.

En general, ya se ha mostrado como el sistema es capaz de contrastar las normas creadas contrastando normas dadas por el CIP en la sección 4.1.2 y su exactitud depende de cómo se creen los archivos que contengan las reglas. Idealmente cada archivo debe contrastar una regla específica, un conjunto de reglas relacionadas o extractos independientes de una misma regla.

El flujo de trabajo del software se divide en dos componentes principales:

1. Procesamiento de las reglas:
 - Se define una sintaxis específica para escribir las reglas que se desean evaluar.
 - Las reglas se componen de instrucciones que establecen relaciones entre objetos, definen cláusulas condicionales y especifican acciones a realizar.
 - Se utilizan técnicas de análisis léxico y sintáctico para procesar las reglas y generar una estructura de datos que representa la lógica de evaluación.
2. Evaluación del modelo BIM:
 - Se interactúa con la biblioteca *ifcopenshell* para leer y extraer la información geométrica del modelo BIM almacenado en un archivo IFC.
 - Se utilizan algoritmos de procesamiento geométrico para calcular propiedades espaciales relevantes, como alturas, distancias, intersecciones y relaciones entre elementos arquitectónicos.

Una vez que se han procesado las reglas y se ha extraído la geometría del modelo BIM, el software combina ambas partes para realizar la evaluación de cumplimiento. Utiliza la estructura de datos generada a partir de las reglas y aplica los cálculos geométricos necesarios sobre los elementos del modelo. El resultado de esta evaluación indica si el modelo cumple o no con las reglas establecidas.

4.2.3 EJECUCIÓN AUTOMÁTICA Y SALIDA DEL PROGRAMA

Las respuestas obtenidas por el prototipo, se dividen en tres métodos de visualización diferentes, en caso que exista uno o más elementos que en su interacción incumpla con la norma, dos consisten en la visualización de las representaciones tridimensionales de los elementos requeridos en la formación de interacciones de cada regla, mientras que el tercero consiste en una impresión en el terminal de ejecución que indica cada fila de la matriz de evaluación que tenga un valor falso, es decir, los elementos que en su interacción incumple la norma descrita y los valores verdadero\falso sobre cada cláusula que conforme la regla.

1. Primer Visualizador: Objetos que interactúan.

El primer visualizador mostrará todos los elementos que en su interacción produce el incumplimiento de la norma, donde cada tipo de elemento tendrá un color particular. Un ejemplo:

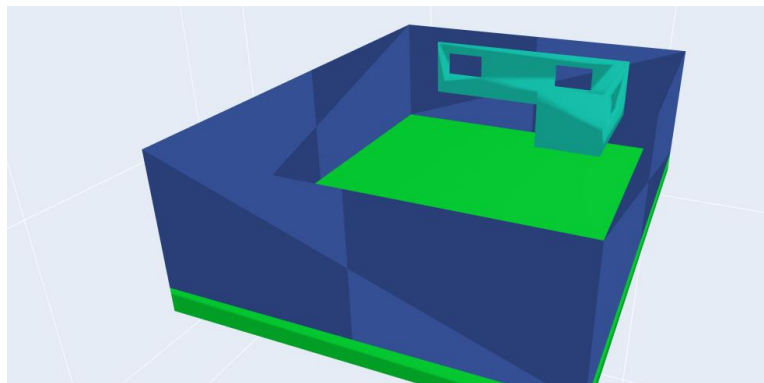


Figura 24: Ejemplo Visualizador elementos que incumplen normas de distanciamiento

Fuente: Elaboración Propia

Donde los elementos de color verde representan los elementos tipo terreno, los celestes los muros que incumplen las normas y en azul los deslindes de la propiedad.

2. Segundo Visualizador: Objetos que interactúan vs objetos que no interactúan:

El segundo visualizador mostrará todos los elementos del proyecto necesarios para la producción de la regla analizada, se mostrarán en color azul todo elemento que en su interacción produce el no incumplimiento de la norma analizada y en color amarillo aquellos que sí producen el incumplimiento de la norma. Todos los objetos mostrados en el primer visualizador aparecerán coloreados amarillos en el segundo visualizador.

Por ejemplo:

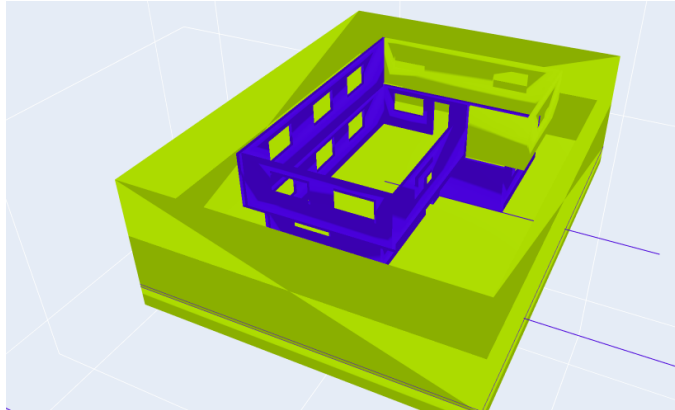


Figura 25: Ejemplo Visualizador elementos que incumplen normas con referencia de objetos que sí cumplen.

Fuente: Elaboración propia

3. Impresión en el terminal de ejecución:

Indicará cada fila de la matriz de evaluación que tenga un valor falso, es decir, los elementos que en su interacción incumple la norma descrita y los valores verdadero\falso sobre cada cláusula que conforme la regla. Ejemplo:

```
MUROS:
(
  Representacion Geometrica
)
PLANO_DESLINDES:
(
  Representacion Geometrica
)
TERRENO:
(
  Representacion Geometrica
)
CLAUSULA 1:
False
CLAUSULA 2:
False
CLAUSULA 3:
True
CLAUSULA 4:
True
CLAUSULA 5:
False
CLAUSULA 6:
True
CLAUSULA 7:
True
RESPUESTA:
(False, 0)
```

Figura 26: Ejemplo fila de matriz evaluación cuya interacción entre los objetos resultan en incumplimiento de la norma

Fuente: Elaboración propia

Este visualizador permite observar con mayor detalle a que cláusula se debe el incumplimiento respecto a que objetos. Dado todo lo anterior, se valida que el prototipo creado tiene la capacidad de entregar como respuesta, tanto un listado de observaciones del plano en un lenguaje que aporte al arquitecto proyectista la posibilidad de corregir su propuesta, con conocimiento de causa como la capacidad de identificar visualmente los elementos que incumplen con una normativa.

Respecto a la automatización, excluyendo que la creación de reglas depende de la programación en un DSL para crearlas, una vez hecho ello permite la automatización de cualquier proyecto siendo que el flujo donde interactúa un usuario es solo la entrega del proyecto, elección de la regla a evaluar y recibir la respuesta que determina cumplimiento o incumplimiento de esta.

4.3 VALIDACIÓN CON PRUEBA DE CONCEPTOS:

Con tal de validar el sistema, se ha creado una prueba de conceptos con un profesional arquitecto con vasto conocimiento en las reglamentaciones necesarias para aprobar edificaciones en Chile. Esta prueba de conceptos, busca replicar una interacción con el motor bajo la suposición de una interfaz diseñada, donde el realizador de la prueba supliría la ausencia de una interfaz amigable con el usuario. El objetivo principal de esta prueba es demostrar la viabilidad de los conceptos descritos en los objetivos específicos, a través de la evaluación práctica y exhaustiva de la solución previamente propuesta.

La validación con pruebas de conceptos es un instrumento clave en la investigación y desarrollo de tecnologías innovadoras. Esta metodología permite evaluar la funcionalidad y eficacia del sistema en un entorno controlado, anterior a su implementación en escenarios reales.

La metodología de la realización fue una prueba de usabilidad y una entrevista, de forma que se obtuvieron datos cuantitativos y cualitativos. Se presentó un escenario donde el usuario debía crear ocho proyectos con libre disposición in situ en total, dos para cada una de las cuatro reglas formuladas en este primer acercamiento del prototipo, donde de esos dos uno corresponde a uno que cumple la norma y otro donde incumple la norma en cuestión. Estos proyectos fueron creados con el programa Revit 2025, entregados al realizador para posteriormente exportarlo en formato IFC, luego, en una reunión presencial se le explica al usuario las salidas que obtendrá del programa, de modo que pudiese interpretar correctamente lo que observaría de las salidas, por ello se ejecutó individualmente cada proyecto junto al usuario de modo que indicaba la eficacia del prototipo respecto a las metas propuestas. Finalmente, se planteó una entrevista donde se discutió distintas cuestiones de la experiencia del usuario respecto a las capacidad actual y potencial de futuras funcionalidades.

4.3.1 RESULTADOS CUANTITATIVOS:

Se recopilaron datos sobre la precisión, eficiencia y tasa de detección de errores del software. Estos resultados se analizaron para determinar la viabilidad técnica del sistema en la interpretación y verificación de proyectos arquitectónicos.

Se resume los resultados de la evaluación en la siguiente tabla, donde se consideran los siguientes parámetros:

1. Responde correctamente si cumple o no proyecto respecto norma evaluada:

Si el proyecto creado para cada regla cumple la expectativa del usuario que cumpla o no la regla, entonces responde SI. Por otro lado, si no cumple la expectativa del usuario, y por ende la salida del prototipo erró en su resultado de cumplir o no, entonces responde NO.

2. Muestra correctamente todos los elementos que incumplen norma:

Si los visualizadores muestran correctamente todos los elementos del proyecto que incumplen la norma, en caso de que deban incumplir correctamente o si en caso de cumplir la norma no muestra ningún elemento en los visualizadores, entonces responder SI. Si en cuyo caso de que el proyecto debe incumplir la norma no muestra todos los elementos del proyecto que incumplen la norma en los visualizadores, o en muestra cualquier elemento en situación que el proyecto si cumple la norma, entonces responder NO.

3. Identifica correctamente los elementos referentes al proyecto original:

Es posible identificar las referencias de los elementos dados en los visualizadores en caso de que los proyectos incumplan cualquier norma respecto al proyecto original, entonces responder SI. En el caso que no logre identificar los elementos mostrados en los visualizadores respecto al proyecto original, entonces responder NO.

Nombre proyecto	Responde correctamente si cumple o no proyecto respecto norma evaluada.	Muestra correctamente todos los elementos que incumplen norma.	Identifica correctamente los elementos referentes al proyecto original
Rasante Cumple	SI	-	-
Rasante no Cumple	SI	NO	SI
Distanciamiento Cumple	SI	-	-
Distanciamiento no Cumple	SI	SI	SI
Antejardín Cumple	SI	-	-
Antejardín no Cumple	SI	NO	SI
Altura Máxima Cumple	SI	-	-
Altura Máxima no Cumple	SI	SI	SI

Tabla 27: Resultado prueba conceptos

Fuente: Elaboración propia

Respecto al primer punto de la tabla anterior para todos los casos el prototipo creado, cumple la función de poder distinguir entre aquellos proyectos cuales cumplen o incumplen cada norma con precisión. Donde flaquea el sistema es en la identificación de todos los elementos involucrados en el incumplimiento de una norma. Mas esto se explica por las funciones utilizadas en la declaración de la regla en el DSL, ya que, aunque capaz de identificar correctamente si infringe la norma evaluada, no logra mimetizar el comportamiento del proceso completo de revisión. De forma final, Las salidas fueron suficientes para que el arquitecto fuese capaz de identificar todos los elementos que incumplían las normas a pesar de lo anterior y logró exitosamente identificar en el proyecto Revit original los pasos necesarios para solucionar dichas infracciones.

Los casos donde no logra Mostrar correctamente todos los elementos que incumplen norma son:

1. **Antejardín no Cumple:** El problema se encuentra que no identifica el muro frontal de la propiedad que ocupa el espacio de antejardín, cuando ningún elemento de construcción puede ocupar el antejardín. Este comportamiento se debe a que la función utilizada para identificar si un elemento traspasa el espacio de antejardín es PLANO_ES_SOBREPUESTO, función que solo identifica aquellos elementos del proyecto que interactúan con el eje de la línea de edificación. El comportamiento ideal, realizado por revisores, es identificar todo elemento que traspase dicha línea, no que se sobreponga a ella. La solución seria crear una nueva función con dicho comportamiento, puesto que la función actual actúa bien como *dummie* con tal de identificar si el proyecto interactúa con dicha línea.

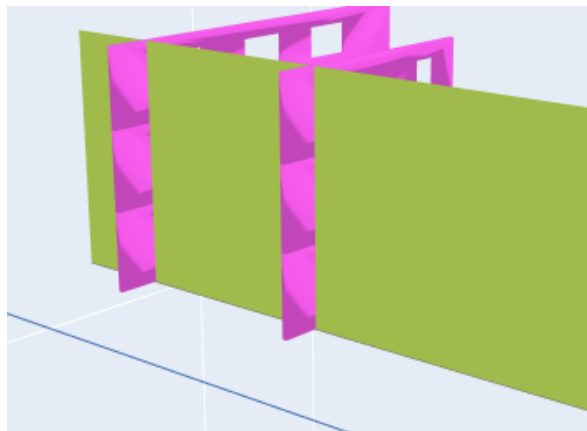


Figura 27: Evaluación de antejardín

Fuente: Elaboración propia

2. **Rasante no cumple:** Similar al caso anterior, el problema se encuentra que no identifica el muro frontal de la propiedad que traspasa el eje de la línea de edificación. Este comportamiento se debe a que la función utilizada para identificar si un elemento traspasa las rasantes es PLANO_ES_SOBREPUESTO, función que solo identifica aquellos elementos del proyecto que interactúan con cualquiera de las rasantes, que en la **Figura 28** se representan como los elementos color verde. El comportamiento ideal, realizado por revisores, es identificar todo elemento que traspase dicha línea, no que se sobreponga a ella.

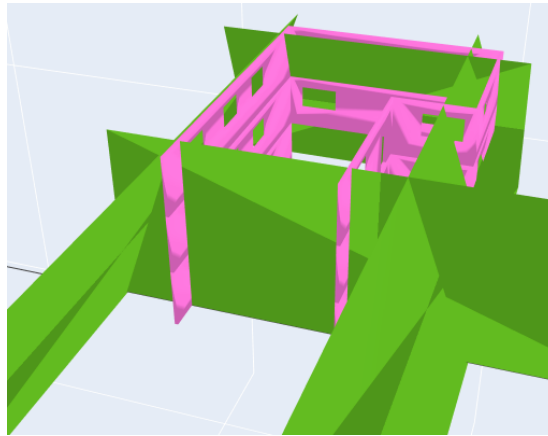


Figura 28: Evaluación Rasantes

Fuente: Elaboración propia

A pesar de los casos donde no logra mostrar correctamente todos los elementos que incumplen, el visualizador tridimensional permitió al arquitecto igualmente identificar que dichos muros faltantes en ambos casos estaban incumpliendo la norma.

4.3.2 RESULTADOS CUALITATIVOS:

Se obtuvo retroalimentación del arquitecto evaluador sobre la usabilidad, claridad de los informes generados y la utilidad del sistema en la práctica profesional. Esta información fue crucial para identificar áreas de mejora y validar la hipótesis de que el sistema puede facilitar la revisión normativa y reducir los tiempos de aprobación de proyectos.

En la entrevista se realizaron 5 preguntas de retroalimentación que permiten identificar la fluidez, confiabilidad, usabilidad, percepción y posibles mejoras. Las respuestas específicas se encuentran en el **Anexo A**.

En primer lugar, el usuario genera altas expectativas a lo que puede lograr el prototipo creado, seguido de una decepción respecto al alcance y funcionalidades actuales que se logran, para de forma final aceptar el estado actual de desarrollo y generar una discusión del potencial de los conceptos a demostrar en esta prueba. El usuario termina el experimento cambiando su percepción, desarrollando altas expectativas respecto a mejoras de la interfaz de usuarios de las salidas y de posibles utilidades en distintos contextos del desarrollo de un proyecto de arquitectura y construcción, como puede ser el potenciar herramientas para Revit, aplicaciones en páginas de DOM, etc.

En general, el usuario logra entregar fluidamente su percepción del programa entregando una retroalimentación sustancialmente positiva para las intenciones de desarrollo de los conceptos demostrados.

Los resultados de la entrevista indica que, excluyendo su interacción indirecta con el prototipo para cargar los proyectos, el usuario consideró como fluida e intuitiva al poder permitirle con un único vistazo reconocer los errores que están ocurriendo y fácil de identificar cuando un proyecto es exitoso. Aunque no existen herramientas que rectifiquen con nombre y sentido de cada elemento, ya que la identificación se realiza a través de la representación geométrica del elemento, la interfaz visual resultó suficiente para entender lo sucedido. Además, reconoce con de fácil lectura la lógica de cláusulas explicadas en la sección **3.11.1**. Por otra parte, logra identificar que los criterios de las cláusulas de las reglas coinciden con los utilizados por él al momento de realizar una revisión manual de cualquiera de las normativas, agregando que el resultado es inequívoco.

Respecto a la usabilidad del programa respecto a sus actividades profesionales, responde con su preferencia de incluir una esta aplicación directamente un el programa licenciado de Revit.

Referente a las funcionalidades adicionales o cambios sugeridos, busca que se siga desarrollando funciones más complejas para más normas y reglas. Aunque reconoce que el prototipo es expandible gracias a su DSL y que mejoraría la revisión de reglas exactas, las cuales muchas veces dan pie a malas interpretaciones de revisores cuyas decisiones suelen atrasar los procesos de revisión de proyectos.

CAPÍTULO 5: CONCLUSIONES

El presente trabajo se ha centrado en el desarrollo y validación de una solución automatizada para la revisión de planos arquitectónicos, utilizando tecnologías de Arquitectura 4.0 y Modelado de Información de Construcción (BIM). Este proyecto surgió de la necesidad de abordar los desafíos existentes en el sistema público de revisión de proyectos arquitectónicos, el cual actualmente presenta cuellos de botella significativos que impactan negativamente en la eficiencia y precisión de los procesos de verificación normativa. La creciente complejidad de los proyectos arquitectónicos y la diversidad de normativas locales hacen que la revisión manual sea cada vez más susceptible a errores y demoras.

Se ha identificado que una de las principales causas de los retrasos en la obtención de permisos de edificación y recepción final de obras es la falta de uniformidad en la interpretación y aplicación de las normativas urbanísticas. Esta falta de uniformidad no solo genera incertidumbre entre los profesionales de la construcción, sino que también incrementa los costos y tiempos asociados a la corrección y reenvío de documentos.

El enfoque de esta investigación se ha basado en la implementación de un motor de verificación automatizado que utiliza modelos BIM para analizar y verificar el cumplimiento de las normativas urbanísticas.

La metodología empleada en esta tesis ha incluido una serie de etapas clave, comenzando con la identificación de los requisitos normativos y técnicos, el desarrollo del software de verificación, y la realización de pruebas de concepto. Estas pruebas han permitido validar la viabilidad técnica de la solución y evaluar su desempeño en condiciones controladas.

Los resultados obtenidos de esta investigación han demostrado que la automatización de la revisión de planos arquitectónicos puede reducir significativamente los tiempos de tramitación y mejorar la precisión de las evaluaciones normativas. Sumo a esto, la implementación de esta tecnología puede contribuir a una mayor transparencia y equidad en el proceso de aprobación de proyectos, beneficiando tanto a los profesionales de la construcción como a las autoridades reguladoras.

5.1 RESUMEN DE LOS OBJETIVOS ALCANZADOS

En primer lugar, se identificaron las principales limitaciones y retos en el sistema actual de revisión de proyectos arquitectónicos, especialmente los retrasos y la falta de uniformidad en los criterios aplicados por las DOM. Este análisis inicial permitió establecer una base sólida para el desarrollo de una solución que abordara estos problemas de manera efectiva.

El primer objetivo fue desarrollar un software capaz de interpretar planos arquitectónicos digitales y verificar su cumplimiento con las normativas urbanísticas. El prototipo desarrollado fue sometido a una serie de pruebas de concepto que demostraron su capacidad para realizar estas tareas.

El segundo objetivo se centró en la creación de un motor de verificación automatizado que pudiera aplicar las reglas normativas de manera consistente y uniforme. Este motor fue diseñado para no solo comparar los datos extraídos de los modelos BIM con las normativas establecidas, generando informes detallados sobre el cumplimiento normativo de cada proyecto, sino que también se ha creado con la idea de poder seguir generando y corrigiendo las metodologías de revisión con un DSL sin tener que entender netamente programación, de modo que sea escalable tanto a nivel nacional como nivel municipal, tal que se puedan fácilmente modificar según las características de cada localidad. Durante el proceso de validación, se comprobó que el motor podía identificar correctamente los incumplimientos normativos y ofrecer recomendaciones específicas para su corrección, lo cual representa una mejora significativa en comparación con los métodos manuales tradicionales.

Otro aspecto clave del proyecto fue la interacción con un profesional arquitecto para validar la usabilidad y efectividad del sistema. A través de pruebas prácticas, se evaluó cómo el sistema podría integrarse en el flujo de trabajo de un arquitecto, facilitando la revisión de proyectos y reduciendo los tiempos de tramitación. La retroalimentación obtenida de estas pruebas fue crucial para ajustar y perfeccionar el prototipo, asegurando que cumpliera con las expectativas y necesidades del usuario final. Adicionalmente, se consideraron escenarios variados para evaluar la adaptabilidad del sistema a diferentes tipos de proyectos y normativas.

El alcance del proyecto incluyó la creación de una interfaz de usuario que, aunque básica en esta etapa inicial, permitió demostrar la viabilidad de una solución más integrada y amigable para el usuario objetivo. Este desarrollo sentó las bases para futuras mejoras y expansiones del sistema, enfocadas en optimizar la experiencia del usuario y aumentar la funcionalidad del software.

En resumen, se logró cumplir con los objetivos establecidos al inicio de este proyecto, desarrollando un prototipo funcional que mejora significativamente la eficiencia y precisión en la revisión de proyectos arquitectónicos. Los resultados obtenidos validan la hipótesis de que una solución automatizada basada en tecnologías de Arquitectura 4.0 y BIM puede transformar el proceso de verificación normativa, beneficiando a los profesionales de la construcción y las entidades reguladoras por igual. Este trabajo no solo proporciona una solución práctica a un problema real, sino que también abre la puerta a futuras investigaciones y desarrollos en el ámbito de la digitalización de procesos constructivos.

5.2 PROGRESOS DE LA “PERMISOLOGÍA”:

La “permisología” en Chile, entendida como el conjunto de trámites y procesos necesarios para obtener permisos de edificación, se enfrenta a desafíos significativos, siendo uno de los más críticos la falta de competencia de los profesionales municipales en la revisión y aprobación de proyectos arquitectónicos. Este proceso, gestionado principalmente por las DOM, está plagado de retrasos debido a la complejidad y variabilidad de las normativas urbanísticas.

Irónicamente cada municipio puede tener interpretaciones distintas de las normas, lo que lleva a inconsistencias en la aplicación de las normativas y, en consecuencia, a tiempos prolongados de tramitación. Las consecuencias de estos problemas son múltiples. Los retrasos en la obtención de permisos de edificación pueden generar costos adicionales significativos para los desarrolladores, quienes deben esperar largos periodos antes de comenzar la construcción. Además, la falta de uniformidad en la interpretación de las normas puede dar lugar a decisiones arbitrarias, afectando la transparencia y equidad del proceso. Este escenario no solo desalienta la inversión en proyectos de construcción, sino que también puede retrasar el desarrollo urbano y la disponibilidad de viviendas y otros edificios necesarios para la comunidad.

En la práctica, los estudios profesionales pueden resultar insuficientes para comprender la magnitud y complejidad de todos los procesos necesarios para obtener permisos. Esta situación puede dar lugar a revisores ineptos, cuya falta de pericia y conocimiento adecuado de las normativas pueden comprometer la calidad de las evaluaciones. Por otro lado, esta falta de uniformidad y preparación puede abrir la puerta a prácticas poco éticas, corruptas e incompetentes, pues del mismo modo que en estación central se permitió la construcción de guetos verticales que claramente incumplen cualquier norma, donde las viviendas contravienen cualquier estándar de dignidad humana en pos del lucro privado.

Conscientes de estos problemas, los gobiernos recientes han implementado varios esfuerzos para digitalizar los procesos de revisión y aprobación de proyectos. Uno de los proyectos más destacados en este ámbito es el "Plan BIM", lanzado por el Ministerio de Obras Públicas. Este plan tiene como objetivo promover la adopción del BIM en los proyectos públicos para mejorar la eficiencia y la transparencia en la construcción. Este plan se estructura en varias fases que incluyen la capacitación de profesionales, la creación de estándares nacionales para la implementación de BIM y la digitalización de procesos clave en la construcción. Otro proyecto importante es el "Programa de Modernización de la Dirección de Obras Municipales", que busca modernizar y digitalizar las funciones de las DOM para mejorar la gestión y revisión de proyectos.

En este contexto, el motor de revisión automática diseñado en esta tesis puede jugar un papel crucial en potenciar estos esfuerzos de digitalización. La capacidad del software para interpretar modelos BIM y verificar automáticamente el cumplimiento normativo, representa un avance significativo hacia la estandarización y eficiencia de los procesos de revisión. Este motor no solo reduce el tiempo necesario para la aprobación de proyectos, sino que también asegura una aplicación uniforme de las normativas, eliminando la subjetividad y variabilidad en la interpretación de las reglas. Adicionalmente, considerando que el objetivo de este motor no es comprometerse a ninguna interfaz partícula, ello puede permitir potenciar distintas aplicaciones de los nuevos procesos de digitalización, aplicándose como módulos de librerías de Revit, revisores online, un paso previo al envío de proyectos a los DOM con tal de servir como un filtro rápido de identificación de errores o aplicaciones que permitan simplificar estudios de cabida en todo un sector urbano.

La aplicación de reglas normativas mediante algoritmos reduce el margen de error humano, asegurando que todas las normativas relevantes se cumplan de manera precisa. La estandarización de la interpretación y aplicación de las normativas urbanísticas garantiza decisiones más justas y transparentes. Asimismo, la generación automática de informes detallados sobre el cumplimiento normativo facilita el monitoreo y control por parte de las autoridades reguladoras, mejorando la transparencia del proceso.

Es menester mencionar que este programa no debería ser utilizado para remplazar totalmente el deber de los revisores, arquitectos o ingenieros dedicados a la “permisología”, sino que busca mejorar sus procesos eficiente y eficazmente. Ya que siempre se requerirá una aprobación humana de modo que siga existiendo personal responsable de cada proyecto en casos de reclamaciones, errores, etc.

A modo de conclusión final, la integración de un motor de revisión automática en el proceso de “permisología” no solo alinea con los esfuerzos gubernamentales de digitalización, sino que también ofrece una solución práctica y eficaz para los desafíos actuales en la revisión de proyectos arquitectónicos. Al adoptar esta tecnología, Chile puede avanzar significativamente hacia una gestión más eficiente y transparente de sus proyectos de construcción, beneficiando a desarrolladores, autoridades, comunidades y civiles por igual, de modo que se permita aumentar el flujo de proyectos que se puedan revisar en un momento dado y dando la oportunidad de dar prioridades a proyectos no convencionales o de mayor envergadura. El futuro de la construcción en Chile se vislumbra prometedor, con procesos más ágiles, precisos y justos, impulsados por la innovación tecnológica y la digitalización. La implementación de un motor de revisión automática no solo mejora la eficiencia y precisión en la revisión de proyectos, sino que también contribuye a reducir la corrupción y aumentar la transparencia, garantizando un desarrollo urbano más equitativo y sostenible.

5.3 TRABAJO FUTURO:

Los trabajos futuros deben centrarse en metodologías que permitan traducir las normas de modo automático en un lenguaje fácilmente comprensible respecto a un DSL, determinando métodos primarios para determinar cada norma computable y metodologías de clasificación de las normas de modo que se puedan organizar de modo que emule todos los reglamentos. Lo ideal es que para cada nueva norma existente se debería crear su versión digital de modo que cualquier aplicación potenciada con un motor de revisión de planos digitales pueda interpretarlo.

Otra prioridad, es seguir desarrollando dicho motor como código no licenciado y abierto, pues debe ser prioritario la transparencia de cada método.

Ya que, aunque ya existen programas que permitan automatizar reglas, es necesario que un set de herramientas, como el diseñado en este documento, permita potenciar el trabajo público y privado por igual.

5.4 PALABRAS FINALES DEL AUTOR:

Este trabajo ha sido un enorme desafío personal, donde he cruzado por varios baches en mi vida persona que han hecho esta experiencia más difícil de lo usual. Este proyecto no solo es el final de los estudios que he realizado, sino que es el final de una era de mi vida y como tal, el terror a la incertidumbre de una nueva era ha sido un fantasma que ha pesado en mi espalda, pero aun así y gracias a todos aquellos que confiaron en mi capacidad, logré salir adelante con un trabajo del cual me siento orgulloso de haber terminado. Creo que este trabajo podría abrir las puertas a nuevos mercados y mejoras en los sistemas actuales.

Mi padre es arquitecto, siempre lo he acompañado a sus reuniones con clientes, viajes a terreno y visitas a municipalidades. He vivido escuchando sus quejas de la ineficiencia del modelo actual y mi motivación en este trabajo es poder crear algo de lo cual pueda no solo ayudarlo a él, sino que crear algo que mejore las posibilidades de que cualquiera pueda construir su hogar con profesionales, pues puede llegar a ser decepcionante intentar construir un hogar y verse limitado por burocracia o carencias de entendimiento de las normas de un arquitecto.

Este trabajo busca sobre todo mejorar la transparencia y acercar la automatización a todos por igual.

Me gustaría también añadir la importancia de la salud mental, ya que tuve que enfrentar problemáticas a nivel psicológico que me hicieron posponer el trabajo de tesis más de lo usual, sin embargo, rescaté y aprendí de la experiencia como es de beneficioso tener apoyo médico cuando uno cursa una enfermedad psicológica, puesto que son invisibles al ojo humano, pero la sintomatología existe y es real.

ANEXOS

Anexo A ACTA PRUEBA DE CONCEPTO

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE**



Acta prueba de concepto:

**“Prototipo de interprete digital de planos
arquitectónicos para revisión automatizada
respecto a regularización de construcción
chilena”.**

Realizador:

Felipe Alberto Enrique Narea Gago

Evaluador:

Jorge Alberto Narea Toro.

Junio – 2024

Introducción:

Con la finalidad de mejorar los procesos actuales de los servicios de revisión de proyectos de las Direcciones de Obras Municipales de Chile, se ha creado un prototipo de revisor automático, cuyo objetivo es complementar los actuales protocolos de los revisores municipales, asimismo asistir a revisores independientes y arquitectos. Este prototipo actual, tiene el alcance e intención de demostrar el concepto de un “intérprete digital de planos arquitectónicos para revisión automatizada respecto a regularización de construcción chilena”, el cual cuenta con capacidad de recibir proyectos arquitectónicos en formato IFC de cualquier programa de arquitectura que trabaje con sistema BIM e interpretar un lenguaje de dominio específico diseñado para describir reglas y normativas, con el fin de retornar si para un proyecto dado cumple o no una norma.

Este lenguaje de dominio específico declara reglas en cláusulas y secciones de cláusulas respetando la lógica proposicional, donde se entenderá en general cada cláusula como una condición elemental e indivisible de la regla, y se dará a entender una sección de cláusulas como secciones condicionales distintas de la propia regla.

Este revisor deberá entregar como respuesta un valor verdadero en consola en caso de que el proyecto cumpla la regla analizada. En caso contrario, retornará un visualizador que mostrará todos aquellos elementos que interactúan y provocan el incumplimiento de la regla y una matriz que indica con mayor precisión la interacción entre objetos.

En esta evaluación se consultará a un usuario experto en la materia su experiencia de uso y se discutirá el potencial y atractivo del prototipo en situaciones reales de uso.

Objetivo Generales:

Instar al usuario experto a crear proyectos arquitectónicos en un programa de metodología BIM con libre disposición, con tal que algunos posean errores intencionales, además de probar si el prototipo puede identificar correctamente dichos errores y reportarlos de manera comprensible para el usuario. Posteriormente se propondrá una instancia de retroalimentación para discutir el potencial de la herramienta, posibles mejoras y desventajas.

Este prototipo alcanza un TRL 3 – “Prueba experimental de concepto”, en la escala de nivel de madurez tecnológica en el estándar de la UE. Se espera demostrar los conceptos de procesar proyectos arquitectónicos en un formato estándar con tal de contrastarlos con las normas nacionales de construcción, y de entregar un listado de observaciones del proyecto arquitectónico en un lenguaje visual que aporte al arquitecto-revisor la posibilidad de corregir su propuesta con conocimiento de causa

Objetivos Específicos:

1. Crear un conjunto de proyectos de vivienda simples, algunos de los cuales deben poseer errores intencionales respecto a las normativas seleccionadas.
2. Ejecutar para cada regla un conjunto de los proyectos creados.
3. Identificar si el prototipo ha detectado los errores correctamente.
4. Discutir la percepción del usuario sobre el programa.
5. Entregar una retroalimentación al realizador de esta evaluación.

Metodología de evaluación:

Perfil de usuario: El usuario debe contar con un título profesional o técnico asociado a la industria de construcción, ingeniería o arquitectura. Debe contar con experiencia comprobable de haber cursado el proceso de evaluación de un proyecto en cualquier municipalidad del país.

Prueba de concepto del prototipo: Se implementa una evaluación a la viabilidad de los conceptos de un nuevo conjunto de herramientas para crear un lenguaje de dominio específico para la formulación de normativas y reglamentos del rubro de arquitectura, ingeniería y construcción, que permita automatizar parcialmente la revisión de un proyecto arquitectónico de una vivienda unifamiliar y que entregue un listado visual de las observaciones respecto al cumplimiento o no de las normas.

Con miras a demostrar la usabilidad actual de las herramientas desarrolladas y potencial para mejorar las metodologías actuales de revisión de proyectos manuales.

Prueba de usabilidad: La evaluación se centra en observar a los usuarios interactuando con el sistema desarrollado.

Durante la sesión, el participante realizará una serie de tareas y escenarios representativos, diseñados para evaluar el cumplimiento de los requisitos y la experiencia del usuario. Mientras los participantes interactúan con el sistema, el realizador observará y recopilará datos sobre:

- Facilidad de navegación y comprensión de las funcionalidades.
- Tiempo y esfuerzo requeridos para completar las tareas.
- Dificultades, dudas o frustraciones experimentadas por los usuarios.
- Sugerencias y recomendaciones de mejora aportadas por los participantes.

Entrevista: Se llevará a cabo una entrevista semiestructurada con cada participante al finalizar la sesión de evaluación. La entrevista tiene como objetivo profundizar en la percepción y experiencia de los usuarios con el programa.

Durante la entrevista, se abordarán preguntas relacionadas con:

- Utilidad percibida del programa para la validación de proyectos de construcción.
- Facilidad de uso y adaptación a los flujos de trabajo de los usuarios.
- Precisión y confiabilidad de los resultados generados por el programa.

- Funcionalidades más útiles y áreas de mejora identificadas por los usuarios.
- Sugerencias y recomendaciones para futuras versiones o desarrollos.

Interacción esperada del usuario.

El evaluador tomará el rol de usuario que simule ser un revisor independiente que desea revisar varios proyectos respecto a varias reglas de la Ordenanza General de Urbanismo y Construcciones. Este tendrá que interactuar con el realizador, quien tomará el rol de interfaz para interactuar con el prototipo, de modo que pueda entregar los proyectos arquitectónicos según indiquen las instrucciones e interactuar con las salidas de cada evaluación. El evaluador o usuario no interactuará directamente con el sistema.

Instrucciones:

Las reglas por evaluar son las siguientes:

Regla	Descripción
Altura máxima de edificación Art. 2.1.23 OGUC	Se describe en la OGUC como la altura máxima a la cual se puede construir en un terreno respecto a nivel natural del terreno.
Distanciamientos a deslindes Art. 2.6.3 OGUC	Se describe en el OGUC como la distancia mínima que debe existir entre los deslindes de la propiedad y los muros de la construcción, depende de la altura del piso y si los muros poseen vanos o no.
Rasantes de propiedad Art. 2.6.2 y 2.6.3 OGUC	Se describe en el OGUC como planos inclinados con respecto al plano horizontal que se levantan desde el nivel del terreno en todos y cada uno de los puntos que forman los deslindes y el eje de la calle que enfrenta el predio. Donde no se podrá construir sobre las rasantes.
Espacio de antejardín Art. 2.5.8 OGUC	Se describe en el OGUC como el área entre la línea oficial y la línea de edificación, regulada en el instrumento de planificación territorial.

Los pasos por seguir son los siguientes:

1. Se le entregará al usuario una jerarquía de carpetas que describen cada una de las reglas.

2. Para cada regla debe crear al menos 2 proyectos con libre disposición donde uno contenga un incumplimiento con la regla en cuestión y otro que no contenga dicho error. Es necesario almacenar los proyectos en las carpetas correspondiente, debe existir 2 proyectos por carpeta, se permite que en cada regla se utilice el mismo proyecto, pero para cada regla tiene que ser un proyecto distinto.
 - a. Los proyectos deben ser almacenados en formato IFC 4 Reference view con la opción de dividir elementos según nivel.
3. Para cada proyecto debe ejecutar el prototipo y observar la respuesta obtenida indicando si ha obtenido la respuesta esperada en cada ejecución. El realizador registrará los resultados y determinará si la prueba se ha realizado exitosamente o no.
4. Realizado las ejecuciones para las 4 reglas presentadas, se procederá a una entrevista de retroalimentación donde deberá responder a las preguntas dadas por el realizador.

Explicación de Salida:

La respuesta obtenida por el prototipo se divide en tres métodos de visualización diferentes, en caso que exista uno o más elementos que en su interacción incumpla con la norma, dos consisten en la visualización de las representaciones tridimensionales de los elementos requeridos en la formación de interacciones de cada regla, mientras que el tercero consiste en una impresión en el terminal de ejecución que indica cada fila de la matriz de evaluación que tenga un valor falso, es decir, los elementos que en su interacción incumple la norma descrita y los valores verdadero\falso sobre cada cláusula que conforme la regla.

1. Primer Visualizador: Objetos que interactúan.

El primer visualizador mostrará todos los elementos que en su interacción produce el incumplimiento de la norma, donde cada tipo de elemento tendrá un color particular. Un ejemplo:

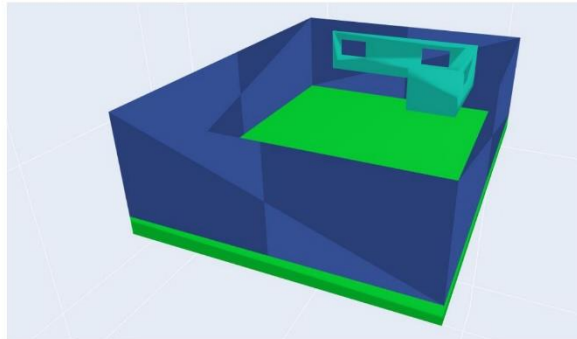


Ilustración 1: Ejemplo Visualizador elementos que incumplen normas

Donde los elementos de color verde representan los elementos tipo terreno, los celestes los muros que incumplen las normas y en azul los deslindes de la propiedad.

2. Segundo Visualizador: Objetos que interactúan vs objetos que no interactúan:

El segundo visualizador mostrará todos los elementos del proyecto necesarios para la producción de la regla analizada, se mostrarán en color azul todo elemento que en su interacción produce el no incumplimiento de la norma analizada y en color amarillo aquellos que sí producen el incumplimiento de la norma. Todos los objetos mostrados en el primer visualizador aparecerán coloreados amarillos en el segundo visualizador. Por ejemplo:

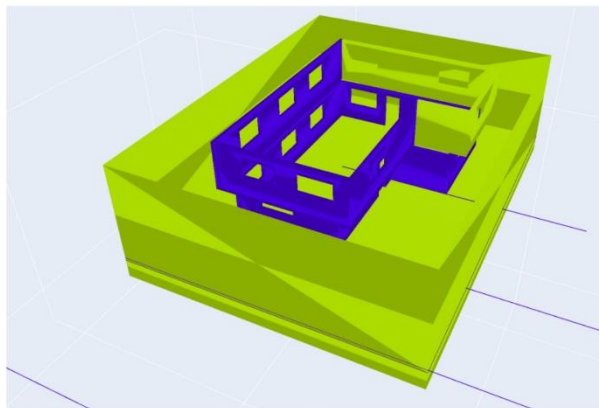


Ilustración 2: Ejemplo Visualizador elementos que incumplen normas con referencia de objetos que sí cumplen.

3. Impresión en el terminal de ejecución:

Indicaré cada fila de la matriz de evaluación que tenga un valor falso, es decir, los elementos que en su interacción incumple la norma descrita y los valores verdadero\falso sobre cada cláusula que conforme la regla. Ejemplo:

```
MUROS:
(
  Representacion Geometrica
)
PLANO_DESLINDES:
(
  Representacion Geometrica
)
TERRENO:
(
  Representacion Geometrica
)
CLAUSULA 1:
False
CLAUSULA 2:
False
CLAUSULA 3:
True
CLAUSULA 4:
True
CLAUSULA 5:
False
CLAUSULA 6:
True
CLAUSULA 7:
True
RESPUESTA:
(False, 0)
```

Ilustración 3: Ejemplo fila de matriz evaluación cuya interacción entre los objetos resultan en incumplimiento de la norma

Resultado de evaluación:

Se resume los resultados de la evaluación en la siguiente tabla, donde se consideran los siguientes parámetros:

1. **Responde correctamente si cumple o no proyecto respecto norma evaluada:**
Si el proyecto creado para cada regla cumple la expectativa del usuario que cumpla o no la regla entonces responde SI. Por otro lado, si no cumple la expectativa del usuario y por ende la salida del prototipo erró en su resultado de cumplir o no, entonces responde NO.
2. **Muestra correctamente todos los elementos que incumplen norma:**

Si los visualizadores muestran correctamente todos los elementos del proyecto que incumplen la norma, en caso de que deban incumplir correctamente o si en caso de cumplir la norma no muestra ningún elemento en los visualizadores, entonces responder SI. Si en cuyo caso de que el proyecto debe incumplir la norma no muestra todos los elementos del proyecto que incumplen la norma en los visualizadores, o en muestra cualquier elemento en situación que el proyecto si cumple la norma, entonces responder NO.

3. Identifica correctamente los elementos referentes al proyecto original:

Es posible identificar las referencias de los elementos dados en los visualizadores en caso de que los proyectos incumplan cualquier norma respecto al proyecto original, entonces responder SI. Caso que no logre identificar los elementos mostrados en los visualizadores respecto al proyecto original, entonces responder NO.

Nombre proyecto	Responde correctamente si cumple o no proyecto respecto norma evaluada.	Muestra correctamente todos los elementos que incumplen norma.	Identifica correctamente los elementos referentes al proyecto original
Rasante Cumple	SI	SI	SI
Rasante no Cumple	SI	SI	SI
Distanciamiento Cumple	SI	SI	SI
Distanciamiento no Cumple	SI	SI	SI
Antejardín Cumple	SI	SI	SI
Antejardín no Cumple	SI	SI	SI
Altura Máxima Cumple	SI	SI	SI
Altura Máxima no Cumple	SI	SI	SI

Análisis y retroalimentación:

Entrevista:

1. ¿Qué tan intuitiva y fluida es la interacción del usuario con el programa?

Respuesta:

La parte grafica es bastante intuitiva y me permite comprende de inmediato los muros los cuales están o no cumpliendo con la norma.

2. ¿Los resultados obtenidos a través del programa son confiables y acordes a las expectativas del usuario?

Respuesta:

Sí, coinciden con el criterio que yo ocupo al revisar un proyecto, solo que esto lo hace más rápido y preciso. El resultado es inequívoco

3. ¿Cómo percibe el usuario la aplicabilidad del programa en sus actividades profesionales?

Respuesta:

Si se pudiera incluir como una aplicación en Revit seria tremendamente útil, para evitar caer en errores.

4. ¿Qué funcionalidades adicionales o cambios sugerirían los usuarios para mejorar el programa?

Respuesta:

Me gustaría que se siguiera desarrollando funciones más complejas como determinación de los conos de sombra o volumen máximo a edificar. Esta situación ha causado innumerables discusiones con las DOM y eventualmente con problemas judiciales posteriores al permiso. También sería ideal superficies según artículo 5.1.11. de la OGUC calcular índice de constructibilidad, ocupación de primer piso y superiores, cantidad de estacionamientos y reducción por bicicletas, distanciamiento de subterráneos a deslindes, etc. En general, las normas urbanísticas contenidas en los Certificados de Informes Previos

Comentarios Adicionales:

Al ser expandible este motor se podrán desarrollar nuevas utilidades que ayuden el trabajo de arquitectos, revisores independientes y municipales. Al tratarse de reglas exactas estás

no dan pie a interpretaciones alambicadas cuya discusión suele atrasar el proceso de revisión del proyecto.

Recomendaciones:

Seguir desarrollando este motor, para alcanzar nuevas utilidades que cubran principalmente aspectos urbanísticos de la OGUC y Planes Reguladores Comunes que son los que en definitiva están bajo la tuición exclusiva de las DOM, y es donde se producen actualmente los cuellos de botella.

Conclusiones:

Respecto a los 5 objetivos específicos descritos anteriormente podemos decir:

1. Crear un conjunto de proyectos de vivienda simples, algunos de los cuales deben poseer errores intencionales respecto a las normativas seleccionadas:

El evaluador crea correctamente los 8 proyectos de vivienda simple según lo indicado en las instrucciones.

2. Ejecutar para cada regla un conjunto de los proyectos creados.

El evaluador con apoyo del realizador logra ejecutar exitosamente el conjunto de los proyectos de viviendas simples creados anteriormente.

3. Identificar si el prototipo ha detectado los errores correctamente:

El evaluador determina que para todos los proyectos creados se ha detectado correctamente si cumplen o no la normas, y en caso de incumplir el prototipo muestra correctamente los elementos que incumplen las condiciones de cada regla.

4. Discutir la percepción del usuario sobre el programa:

En primer lugar, el usuario genera altas expectativas a lo que puede lograr el prototipo creado. Seguido de una decepción respecto al alcance y funcionalidades actuales que se alcanzan, para finalmente aceptar el estado actual de desarrollo y generar una discusión de el potencial de los conceptos a demostrar en esta prueba. El usuario termina el experimento con altas expectativas respecto a mejoras de la interfaz de usuarios de las salidas y de posibles utilidades en distintos contextos del desarrollo de un proyecto de arquitectura y construcción, como puede ser el potenciar herramientas para Revit, aplicaciones en páginas de DOM, etc.

En general, el usuario logra entregar fluidamente su percepción del programa entregando una retroalimentación sustancialmente positiva para las intenciones de desarrollo de los conceptos demostrados.

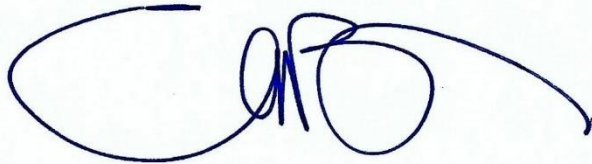
5. Entregar una retroalimentación al realizador de esta evaluación.

Se entrega una retroalimentación en conjunto a la discusión de percepción del usuario sobre el programa, el cual da indicaciones de posibles mejoras a desarrollar en el futuro, nuevas funcionalidades para integración de nuevas reglas y potencial de mercado para un sistema como el desarrollado en este informe.

Perfil Jorge Alberto Narea Toro:

Arquitecto de la Universidad de Chile con vasta experiencia en diseños de centros comerciales, hoteles, edificios de altura, clínicas hospitalarias y viviendas. Por un tiempo fui perito en materias de arquitecturas en las Cortes de Apelaciones de Antofagasta y Valparaíso. Actualmente asesoro a empresas inmobiliarias en la presentación de anteproyectos que permitan alcanzar la máxima constructibilidad y por tanto rentabilidad.

Firma:

A handwritten signature in blue ink, consisting of a large, stylized 'J' followed by a series of loops and a long horizontal stroke extending to the right.

REFERENCIAS BIBLIOGRÁFICAS

[Beach, 2015] Beach, T., et al. (2015). A rule-based semantic approach for automated regulatory compliance in the construction sector. *Expert Systems with Applications*, 42(12), 5219–5231.

[buildingSMART n.d.-a] buildingSMART. (s.f.). Sobre buildingSMART <https://www.buildingsmart.org/about/>

[buildingSMART s.f.-b] buildingSMART. (s.f.). ¿Que es BIM?. <https://www.buildingsmart.es/bim/>

[buildingSMART s.f.-c] buildingSMART. (s.f.). A New Way of Working - buildingSMART Professional Certification. https://education.buildingsmart.org/es/a-new-way-of-working/#es_dim

[buildingSMART s.f.-d] buildingSMART. (s.f.). Industry Foundation Classes (IFC). <https://www.buildingsmart.es/bim/estándares/ifc/>

[Catalogo Arquitectura SpA, 2020] Catalogo Arquitectura SpA. (2020). *OGUC Ilustrada: Vol. I* (Primera edición).

[Construye2025, 2020] Construye2025. (2020). Plan Bin. CONSTRUYE2025. <https://construye2025.cl/iniciativa/plan-bin/>

[Construye2025, 2023] Construye2025. (2023). DOM en Línea analiza los desafíos que impactan en su implementación a nivel nacional. CONSTRUYE2025. <https://construye2025.cl/2023/04/26/dom-en-linea-analiza-los-desafios-que-impactan-en-su-implementacion-a-nivel-nacional/>

[Delis, 1995] Delis, E. A., & Delis, A. (1995). Automatic Fire-Code Checking Using Expert-System Technology. *Journal of Computing in Civil Engineering*, 9(2), 141–156.

[Dimyady, 2016] Dimyadi, et al. (2016). Modelling and Accessing Regulatory Knowledge for Computer-assisted Compliance Audit. *Electronic Journal of Information Technology in Construction*, 22.

[Eastman, 2009] Eastman, et al. (2009). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. John Wiley & Sons.

[Fenves, 1966] Fenves, S. J. (1966). Tabular Decision Logic for Structural Design. *Journal of the Structural Division*, 92(6), 473–490.

[Feito24] Feito, O. B. (2024). ¿Qué son los Objetos BIM y cómo crearlos?. Imasgal. <https://imasgal.com/objetos-bim-y-como-crearlos/>

[Garret, 1987] Garrett, J. H., & Fenves, S. J. (1987). A knowledge-based standards processor for structural component design. *Engineering with Computers*, 2(4), 219–238.

[Herman, 2021] Herman, P. (2021). Falencias en el trámite para obtener permisos de edificación. *Diario y Radio Universidad Chile*.
<https://radio.uchile.cl/2021/02/02/falencias-en-el-tramite-para-obtener-permisos-de-edificacion/>

[Hjelseth, 2010] Hjelseth, E. (2010). EXPLORING SEMANTIC BASED MODEL CHECKING. Norwegian University of Life Sciences.

[LGUC, 1976] Ministerio de Vivienda y Urbanismo. (1976). Ley General de Urbanismo y Construcciones. Decreto con Fuerza de Ley N°458. *Diario Oficial de la República de Chile*, 13 de abril de 1976.

[Lee, 2020] Lee, Y. et al. (2020). A Comparative Analysis of Five Rule-Based Model Checking Platforms.

[Leon, 2023] León, N. (2023). Proyecciones para la construcción. CChC.
<https://www.ccs.cl/wp-content/uploads/2023/11/231123-CChC.pdf>

[Macit, 2017] Macit İlal, S., & Günaydın, H. M. (2017). Computer representation of building codes for automated compliance checking. *Automation in Construction*, 82, 43–58.

[Mercurio, 2022] El Mercurio S. A. P. (2022). Más de un año en promedio: Demoras en permisos de proyectos inmobiliarios explican casi el 10% del precio de viviendas. *Emol*.
<https://www.emol.com/noticias/Economia/2022/12/10/1080684/permisos-proyectos-inmobiliarios.html>

[MINVU, 2016] Ministerio de Vivienda y Urbanismo (MINVU). (2016). Ordenanza General de Urbanismo y Construcciones (OGUC), artículo 2.6.3: Distanciamientos mínimos a los deslindes para construcciones de tipo aislado. Recuperado de <https://www.bcn.cl/leychile/navegar?idNorma=6541>

[MINVU, 2021] Ministerio de Vivienda y Urbanismo (MINVU). (2021). *Manual de solicitante: Modulo de ingresos* [Diapositivas]. [https://providencia.cl/provi/site/docs/20220111/20220111120518/manual del solicitante modulo de ingreso domel v1ago2021.pdf](https://providencia.cl/provi/site/docs/20220111/20220111120518/manual%20del%20solicitante%20modulo%20de%20ingreso%20domel%20v1ago2021.pdf)

[MINVU, 1975] Ministerio de Vivienda y Urbanismo (MINVU). (1975). Ley N°16.391 que crea el Ministerio de Vivienda y Urbanismo. *Diario Oficial de la República de Chile*, 16 de diciembre de 1975.

[MINVU, 1992] Ministerio de Vivienda y Urbanismo (MINVU). (1992). Ordenanza General de Urbanismo y Construcciones. Decreto Supremo N°47. *Diario Oficial de la*

República de Chile, 5 de junio de 1992.
<https://www.bcn.cl/leychile/navegar?idNorma=8201>

[Minvu, 2021] MINVU y CChC impulsan agilización de permisos de construcción. (2021). Inmobiliaria RVC. Retrieved May 20, 2023, from <https://www.rvc.cl/minvu-y-cchc-impulsan-agilizacion-de-permisos-de-construccion/>

[OGUC, 1992] Ministerio de Vivienda y Urbanismo. (1992). Ordenanza General de Urbanismo y Construcciones. Decreto Supremo N°47. Diario Oficial de la República de Chile, 5 de junio de 1992.

[Osorio, 2021] Osorio, V. (2021). Permisos de edificación retroceden y se duplica tiempo de tramitación. Diario Financiero. <https://www.df.cl/empresas/construccion/permisos-de-edificacion-retroceden-y-se-duplica-tiempo-de-tramitacion>

[Pauwels, 2011] Pauwels, P, et al. (2011). A semantic rule checking environment for building performance checking. Automation in Construction, 20(5), 506–518.

[Planbim, 2023] Planbim. (2023). Estándar BIM para proyectos públicos - Planbim. <https://planbim.cl/documentos/estandar-bim-para-proyectos-publicos/>

[Sobhkhiz, 2021] Sobhkhiz, S, et al. (2021). Framing and Evaluating the Best Practices of IFC-Based Automated Rule Checking: A Case Study. Buildings, 11(10), 456.

[Sydora, 2020] Sydora, C., & Stroulia, E. (2020). Rule-based compliance checking and generative design for building interiors using BIM. Automation in Construction, 120, 103368. <https://doi.org/10.1016/j.autcon.2020.103368>

[Tan, 2010] Tan, X., Hammad, A., & Fazio, P. (2010). Automated code compliance checking for building envelope design. Journal of Computing in Civil Engineering, 24(2), 203–211.

[Worldbankgroup, 2024] World Bank Group. (2024). Dealing with Construction Permits - Doing Business. <https://archive.doingbusiness.org/en/data/exploretopics/dealing-with-construction-permits>

[Zhang, 2014] Zhang, C, et al. (2014). Semantic Rule Checking for Regulation Compliance Checking: The Case of the Accessibility of Dutch Buildings. Computer-Aided Civil and Infrastructure Engineering, 29(1), 50-63.

[Zhong, 2012] Zhong, B, et al. (2012). Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking. Automation in Construction, 28, 58–70.