



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Desarrollo de un sistema de información CAD 3D

Proyecto Final de carrera

Autor: Andrés García Morro

Directores: Eduardo Vendrell Vidal

Carlos Sánchez Belenguer

Septiembre de 2013

Introducción	4
Estado del Arte	7
¿Qué es un GIS?	7
Evolución de los GIS	10
GIS 3D.....	13
Aplicaciones.....	21
Aplicación Desarrollada	28
Propuesta de solución.....	28
Herramientas utilizadas	30
Arquitectura de la aplicación	48
Servidor	49
Back-End	52
Front-End	61
Casos de uso	67
Posibles ampliaciones	69
Bibliografía	79

Introducción

El presente proyecto pretende abordar el desarrollo de un sistema de información que permita asociar datos a modelos CAD tridimensionales, estos modelos serán de ámbito arquitectónico y geográfico por lo que se trata de un GIS (Sistema de Información Geográfica). El objetivo final consiste en disponer, por un lado de una herramienta de catalogación y documentación de objetos de naturaleza tridimensional, mayoritariamente edificios e información relacionada y, por otro lado, de una aplicación de visualización de esta información donde además se podrá adquirir nuevos datos gracias al estudio de los modelos cargados y la información almacenada. Con esta herramienta se pretende facilitar el análisis y estudio arquitectónico y geográfico de entidades escaneadas previamente en 3D y introducidas en la base de datos.

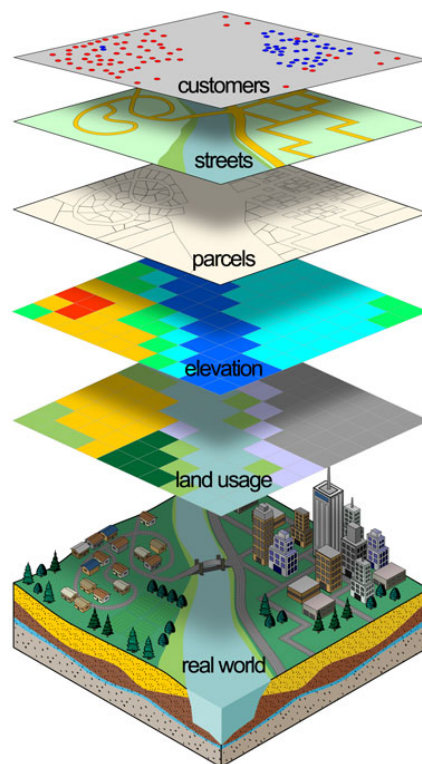


Figura 1: Capas de un GIS (extraída de http://www.gembc.ca/GIS_Mapping_Services_Page.htm)

La visualización de los modelos 3D se llevará a cabo usando el software de simulación 3D en tiempo real Unity 3D. Unity es un motor gráfico centrado en el desarrollo de videojuegos pero que, dada su versatilidad, en los últimos años se ha utilizado para el desarrollo de todo tipo de aplicaciones 3D de ámbito general.

Las principales ventajas de Unity sobre otras tecnologías similares (Virtools, Ogre, OSG...) consisten en su capacidad de publicación multi-plataforma (visor web, aplicación PC/Mac, IOS, Android, Play Station, Wii U y Xbox) y su reducido coste, existiendo una versión gratuita totalmente funcional. Como consecuencia de lo anterior, Unity cuenta con una gran comunidad de desarrolladores y una documentación muy completa y actualizada.

Con el fin de permitir el acceso a los datos desde cualquier ubicación, la base de datos que guarde toda la información se encontrará en un servidor conectado a internet. A esta base de datos se conectarán las dos herramientas: el back-end y el front-end. La primera será la responsable de la gestión de la información, tanto de los modelos como de la información asociada y estará disponible exclusivamente para usuarios con credenciales de "administrador". El usuario accederá con sus credenciales y podrá agregar, modificar o eliminar información, ya sean modelos 3D, archivos PDF, fotografías o datos.

La segunda será disponible para el público en general, y se centrará en la visualización de la información y la de datos gracias al análisis de los modelos y la información alojada en la base de datos, esta parte será implementada con tecnología web, así se podrá acceder desde cualquier localización con acceso a internet.

El objetivo de la separación entre los tres roles (base de datos, back-end y front-end) es que la lógica de la aplicación esté implementada en el gestor de contenidos y en el visualizador, pero que toda la información a mostrar se almacene en una base de datos externa.

Esta separación entre la lógica de visualización y la de almacenamiento de la información en una base de datos accesible desde la red, permitirá la implementación de futuros visualizadores que puedan funcionar sobre plataformas móviles (PDAs, iPads...) y/o visualizadores off-line que puedan distribuirse en USBs/DVDs...

El visualizador a desarrollar adaptará la complejidad de la simulación a los recursos hardware disponible en el equipo de carga, proporcionando resultados de alta calidad en equipos potentes, y resultados correctos en equipos limitados.

Dado que la aplicación desarrollada estará basada en el entorno de red, la arquitectura propuesta para la herramienta de visualización se apoya en el paradigma cliente-servidor con estrategia de entrega de software. De este modo, los servidores no realizan cálculos complejos para cada ejecución individual. En su lugar se envía a los clientes el código de la aplicación para que con, con un player instalado, puedan visualizar los modelos, proporcionando en última instancia la potencia computacional. El empleo de esta estrategia permite que un único servidor proporcione servicios a múltiples clientes, estando únicamente limitado por el rendimiento de su interfaz de red.

La parte para agregar la información en la base de datos permitirá agregar a parte de los modelos 3D, documentos PDF así como fotografías de los modelos y asociarlas a éstos, también se introducirá información respecto a la conservación y construcción de las estructuras. Ésta información también se relacionará con cada parte del modelo 3D como muros, escaleras, ventanas, etc.

Por su parte la web de visualización dispondrá de una parte para navegar por los modelos y acceder a la información relacionada con ellos.

Estado del Arte

¿Qué es un GIS?

Un GIS (Sistema de Información Geográfica) se define como un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente para capturar, almacenar, analizar, transformar y presentar toda la información geográfica y de sus atributos con el fin de satisfacer múltiples propósitos, son una integración organizada de hardware, software y datos geográficos referenciados con el fin de resolver problemas complejos de planificación y gestión geográfica.

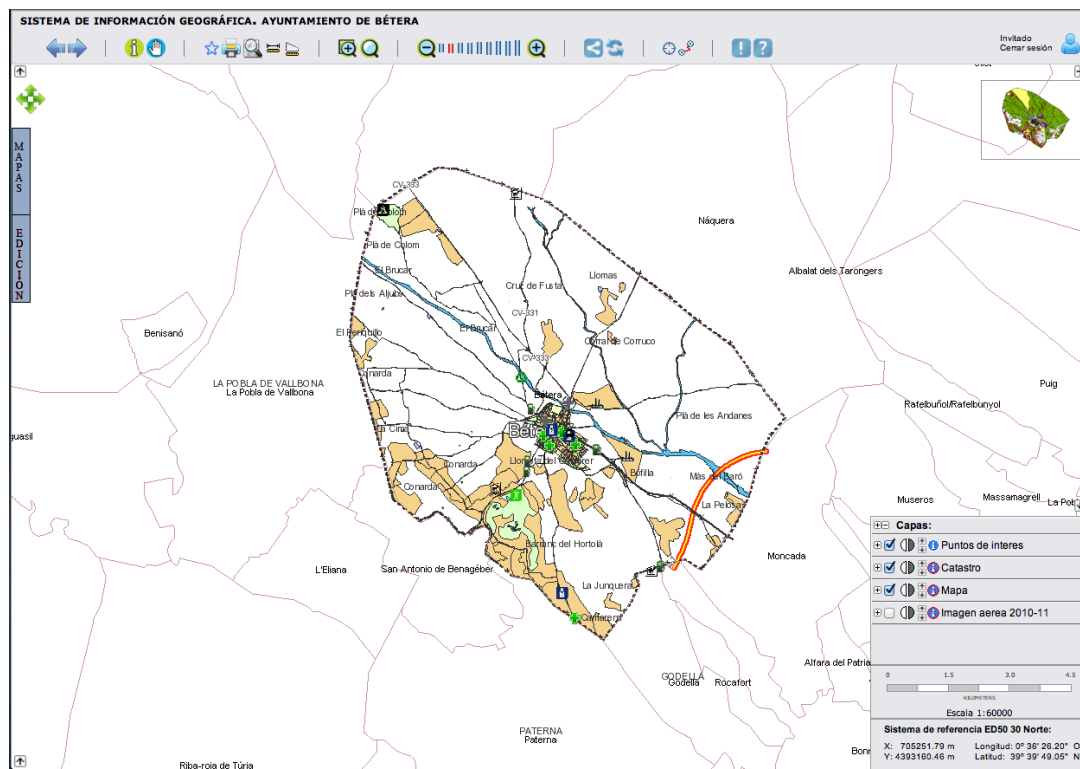


Figura 2: GIS de Bétera (captura de pantalla de <http://sig2.betera.es:8080/visor/>)

En términos prácticos, la función principal de este software es contar con cartografía con bases de datos asociadas, con la misión principal de resolver problemas espaciales o territoriales permitiendo manejar conjuntamente mapas cartográficos y herramientas de estudio de esos mapas.

Las partes en las que se divide un GIS típico son 4, el sistema de manejo de datos, como sería una base de datos típica, con sus consultas y tablas, una interfaz gráfica de usuario para el fácil acceso a las herramientas ya que al tratarse de sistemas que

trabajan con elementos gráficos como los mapas es imposible crear una aplicación de estas características sin ésta parte, también se compone de herramientas para la captura y manejo de información geográfica, esta es la parte más característica de un GIS ya que es la que le otorga la singularidad respecto a otros sistemas de manejo de información, por último están las herramientas para soporte de consultas , análisis y visualización de datos geográficos que es la parte que más diferencia un GIS de otro.

La construcción e implementación de un GIS en cualquier organización es una tarea siempre progresiva, laboriosa y continúa. Los análisis y estudios anteriores a la implantación de un GIS son similares a los que se deben realizar para establecer cualquier otro sistema de información; sin embargo, en los GIS hay que considerar las características especiales de los datos utilizados y sus correspondientes procesos de actualización.

La información geográfica contiene una referencia territorial explícita como latitud y longitud o una referencia implícita como domicilio o código postal.

Los GIS funcionan con dos tipos de diferentes de información geográfica:

- El modelo raster: Basan su funcionalidad en una concepción implícita de las relaciones de vecindad entre los objetos geográficos. Su forma de proceder es dividir la zona de afección de la base de datos en una retícula o malla regular de pequeñas celdas y atribuir un valor numérico a cada celda como representación de su valor temático. Dado que la malla es regular, el tamaño del pixel es constante y se conoce la posición en coordenadas del centro de una de las celdas, se puede decir que todos los pixeles están geo referenciados. Para tener una descripción precisa de los objetos geográficos contenidos en la base de datos el tamaño del pixel debe ser reducido en función de la escala, lo que dotará a la malla de una resolución alta; sin embargo, a mayor número de filas y columnas en la malla, mayor esfuerzo en el proceso de captura de la información y mayor coste computacional al momento de procesarla. El modelo de datos raster es útil cuando tenemos que describir objetos geográficos con límites difusos, como por ejemplo puede ser la dispersión de una nube de contaminantes, o los niveles de contaminación de un acuífero subterráneo, donde los contornos no son absolutamente nítidos; en esos casos, el modelo raster es más apropiado que el vectorial.
- El modelo vector: Son aquellos GIS que para la descripción de los objetos geográficos se utilizan vectores definidos por pares de coordenadas relativas a algún sistema cartográfico. La información sobre puntos, líneas y polígonos se almacena como una colección de coordenadas x, y. La ubicación de una característica puntual, puede describirse con sólo un punto x, y. Las características lineales, pueden almacenarse como un conjunto de puntos de

coordenadas x, y. Las características poligonales, pueden almacenarse como un circuito cerrado de coordenadas.

La construcción de una base de datos geográfica implica un proceso de abstracción para pasar de la complejidad del mundo real a una representación simplificada que puede ser procesada por el lenguaje de las computadoras actuales. Este proceso de abstracción tiene diversos niveles y normalmente comienza con la concepción de la estructura de la base de datos, generalmente en capas; en esta fase, y dependiendo de la utilidad que se vaya a dar a la información a compilar, se seleccionan las capas temáticas a incluir.

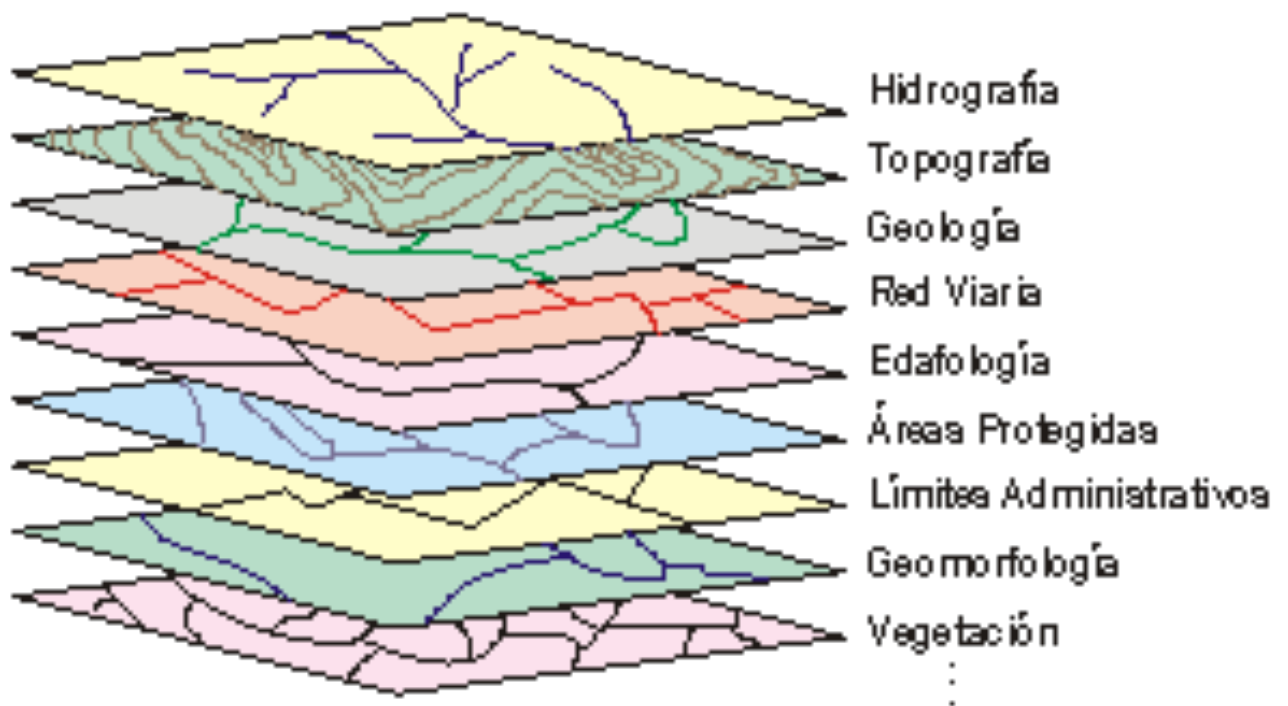


Figura 3: Posibles datos de un GIS (extraída de <http://www.gabrielortiz.com/index.asp?Info=012>)

Pero la estructuración de la información espacial procedente del mundo real en capas conlleva cierto nivel de dificultad. En primer lugar, la necesidad de abstracción que requieren las máquinas implica trabajar con primitivas básicas de dibujo, de tal forma que la complejidad de la realidad ha de ser reducida a puntos, líneas o polígonos.

En segundo lugar, existen relaciones espaciales entre los objetos geográficos que puede llegar a ser muy compleja, ya que son muchos los elementos que interactúan sobre cada aspecto de la realidad.

La topología de un GIS reduce sus funciones a cuestiones mucho más sencillas, como por ejemplo conocer el polígono a que pertenece una determinada línea, o bien saber qué agrupación de líneas forman una determinada carretera.

Existen diversas formas de modelar estas relaciones entre los objetos geográficos o topología. Dependiendo de la forma en que ello se lleve a cabo se tiene uno u otro tipo Sistema de Información Geográfica dentro de una estructura de dos grupos principales: GIS vectoriales y GIS Raster. No existe un modelo de datos que sea superior a otro, sino que cada uno tiene una utilidad específica.

Si nos fijamos en la vida cotidiana con el avance de la tecnología los GIS se han integrado en nuestro día a día. En estos tiempos no es difícil encontrar un coche con GPS (que es un GIS) o usar Google Maps. Para facilitar el acceso a cierta información hay municipios en España que ha creado sus propios sistemas GIS, como Bétera al cual se puede acceder desde esta dirección: <http://sig.betera.es/visor/>, en él, podemos encontrar un mapa de la ciudad así como puntos de interés, el catastro, etc. Otro ejemplo sería el de la comunidad valenciana que se encuentra en <http://www.gvsig.org/web/>.

La Diputación de Valencia está ofreciendo a los municipios de la provincia, la posibilidad de adherirse al proyecto LocalGIS, que tiene por objetivo el desarrollo e implantación de herramientas o soluciones tecnológicas para la modernización de la gestión de la administración local, la idea es implantar extensivamente el proyecto LocalGIS en todos los Ayuntamientos de la provincia de Valencia, especialmente mediante una implantación a nivel provincial. Este proyecto aprovecha la información geográfica existente en fuentes de datos dispersas, permitiendo la interoperabilidad y localización de fuentes de información geográfica proporcionando a los ayuntamientos una herramienta eficaz para la presentación de servicios avanzados de información geográfica.

La tecnología GIS puede ser utilizada para investigaciones científicas, la gestión de los recursos, gestión de activos, la arqueología, la evaluación del impacto ambiental, la planificación urbana, la cartografía, la sociología, etc.

Evolución de los GIS

Hace unos 15.000 años, en las paredes de las cuevas de Lascaux los hombres de Cromañón pintaban en las paredes los animales que cazaban, asociando a estos dibujos unas trazas lineales que, se cree, cuadraban con las rutas de migración de esas especies.

Muchos años más tarde en 1854 el pionero en epidemiología, el doctor John Snow, proporcionaría otro ejemplo cuando cartografió en un mapa la incidencia de los casos de cólera en el distrito de Soho en Londres. Esto permitió al doctor localizar con precisión el pozo de agua contaminado como causante del brote.



Figura 4: Mapa usado por John Snow (extraída de http://es.wikipedia.org/wiki/Sistema_de_Información_Geográfica)

El comienzo del siglo XX vio el desarrollo de la foto litografía donde los mapas eran separados en capas. El avance del hardware impulsado por la investigación en armamento nuclear daría lugar al desarrollo de aplicaciones cartográficas para computadores de propósito general.

El año 1962 vio la primera utilización real de los GIS en el mundo, concretamente en Ottawa y a cargo del Departamento Federal de Silvicultura y Desarrollo Rural. Desarrollado por Roger Tomlinson, fue utilizado para almacenar, analizar y manipular datos recogidos para el Inventario de Tierras Canadá, una iniciativa orientada a la gestión de los vastos recursos naturales del país con información cartográfica relativa a tipos y usos del suelo, agricultura, espacios de recreo, vida silvestre, aves acuáticas y silvicultura, todo ello a escala de 1:50.000. Se añadió, así mismo, un factor de clasificación para permitir el análisis de la información.

El Sistema de Información Geográfica de Canadá fue el primer SIG en el mundo similar a tal y como los conocemos hoy en día, y un considerable avance con respecto a las aplicaciones cartográficas existentes hasta entonces, puesto que permitía superponer capas de información, realizar mediciones y llevar a cabo digitalizaciones y escaneos de datos. Asimismo, soportaba un sistema nacional de coordenadas que abarcaba todo el continente, una codificación de líneas en “arcos” que poseían una verdadera topológica integrada y que almacenaba los atributos de cada elemento y la información sobre su localización en archivos separados. Como consecuencia de esto, Tomlison está considerado como “el padre de los SIG”, en particular por el empleo de información geográfica convergente estructurada en capas, lo que facilita su análisis espacial. El CGIS estuvo operativo hasta la década de los 90 llegando a ser la base de datos sobre recursos del territorio más grande de Canadá. Fue desarrollada como un sistema basado en una computadora central y su fortaleza radicaba en que permitía realizar análisis complejos de conjuntos de datos que abarcaban todo el continente.

En 1964, Howard T. Fisher formó en la Universidad de Harvard el Laboratorio de Computación Gráfica y Análisis Espacial en la Harvard Graduate School of Design, donde se desarrollaron una serie de importantes conceptos teóricos en el manejo de datos espaciales, y en la década de 1970 había difundido código de software y sistemas germinales, tales como SYMAP, GRID y ODYSSEY a universidades, centros de investigación y empresas de todo el mundo.

En la década de los 80, M&S Computing, Environmental Systems Research Institute y CARIS emergerían como proveedores comerciales de software GIS. Incorporaron con éxito muchas de las características de CGIS, combinando el enfoque de primera generación de Sistemas de Información Geográfica relativo a la separación de la información espacial y los atributos de los elementos geográficos representados con un enfoque de segunda generación que organiza y estructura estos atributos en bases de datos.

En la década de los años 70 y principios de los 80 se inició en paralelo el desarrollo de dos sistemas de dominio público. El proyecto Map Overlay and Statical System se inició en 1977 en Fort Collins bajo los auspicios de la Western Energy and Land Use Team y el Servicio de Pesca y Vida Silvestre de Estados Unidos. En 1982 el cuerpo de Ingenieros del Laboratorio de Investigación de Ingeniería de la Construcción del Ejército de los Estados Unidos desarrolla GRASS como herramienta para la supervisión y gestión medioambiental de los territorios bajo administración del Departamento de Defensa.

Esta etapa de desarrollo está caracterizada, en general, por la disminución de la importancia de las iniciativas individuales y un aumento de los intereses a nivel corporativo, especialmente por parte de las instancias gubernamentales y de la administración.

Los 80 y 90 fueron años de fuerte aumento de las empresas que comercializaban estos sistemas, debido al crecimiento de los GIS en estaciones de trabajo UNIX y ordenadores personales. Es el periodo en el que se ha venido a conocer en los GIS como la fase comercial. El interés de las distintas grandes industrias relacionadas directa o indirectamente con los GIS crece en sobremanera debido a la gran avalancha de productos en el mercado informático internacional que hicieron que se generalizase esta tecnología.

En la década de los noventa se inicia una etapa comercial para profesionales, donde los Sistemas de Información Geográfica empezaron a difundirse al nivel de usuario doméstico debido a la generalización de los ordenadores personales o microordenadores.

A finales del siglo XX, principio del XXI el rápido crecimiento en los diferentes sistemas se ha consolidado, restringiéndose a un número relativamente reducido de plataformas. Los usuarios están comenzando a exportar el concepto de visualización de datos GIS a Internet, lo que requiere una estandarización de formato de los datos y de normas de transferencia.

GIS 3D

¿Qué es?

Muchos problemas de GIS solo pueden ser solucionados utilizando 3D. Al visualizar los datos en 3D se pueden resaltar rápidamente las relaciones entre las entidades GIS y las herramientas analíticas pueden cuantificar dichas relaciones en patrones.

Las limitaciones de los sistemas actuales de GIS 2D y de visualización 2.5D – 3D, pueden llevar a plantearnos la pregunta de cómo usar las tecnologías GIS para gestionar la información en edificios urbanos y su entorno; sobre todo en el caso en que se quiera actuar de alguna manera sobre el mismo, tal y como sucede en la rehabilitación del patrimonio en general.

La bibliografía anglosajona suele mencionar las 5 funcionalidades de los GIS, como:

- 1.-Mapping(Cartografiado): El proceso de construcción de mapas siguiendo estándares definidos en naciones o por el usuario.
- 2.-Measurement(Medición): Acción y efecto de medir.
- 3.-Monitoring(Seguimiento): Medidas intermitentes hasta el punto de poder determinar el grado de seguimiento o desviación de una norma determinada.

4.-Modelling(Modelado): El proceso de modelar estructuras.

5.-Management(Gestión): La técnica, práctica o ciencia de gestionar o controlar.

En la actualidad, para elementos 3D, habría que usar ampliaciones sobre los GIS actuales con el fin de que ofreciesen todas las funcionalidades anteriormente citadas de los GIS; ya que estos pierden cierta vigencia, debido a que los sistemas 2.5D-3D deberían cumplir una serie de requisitos para gestionar información. Los datos deberían representar la situación del elemento, preferiblemente en 3D (para introducir información acerca de espesores, interiores a partir de radar gramas, etc.), y la exterior; para esto, los sistemas tradicionales podrían bastar. También debería permitir cambios posibles (edición), y aparecer información del elemento como conductos, escaleras, instalaciones, patologías, etc.

En las últimas décadas, el GIS 3D se ha centrado en la adquisición, visualización, gestión del dato y análisis espacial. Un GIS completamente 3D debería ser capaz de gestionar geometría 3D y topología, integrar dicha geometría e información temática, analizar relaciones espaciales y topológicas, y visualizar datos de una manera apropiada.

En los últimos años la industria se ha centrado en mejorar la captura de los datos en 3D, a través de teledetección, fotogrametría y láser escáner, para visualizarlo bien a través de un CAD o de un software de realidad virtual.

Se han desarrollado muchos modelos en 3D pero muy pocos son apropiados para representar el interior de los edificios y patrimonio en general y su entorno.

Este tipo de información tridimensional de construcciones civiles, es requerida para realizar, por ejemplo, la planeación de la cobertura de las ondas de radio en una población ubicando los rebotes de ondas radiales entre antenas, optimización de redes, ubicación de antenas, interferencias de radio frecuencia, tendido de líneas de transmisión en 3D; o en caso de planificación de un aeropuerto este modelado tridimensional permitiría realizar el estudio de los espacios aéreos que intervienen en el proceso de diseño referenciado, en su caso la viabilidad técnica de su construcción.

Investigaciones

En los últimos años, con el auge de internet y el 3D se ha investigado mucho sobre los GIS y la posibilidad de crearlos en 3 dimensiones para una mejor apreciación de los detalles arquitectónicos. Wang Ming de la Zhejiang University of Media and Communications en su artículo "A 3D web GIS system based on VRML and X3D" hace un estudio de cómo crear un GIS web usando VRML para mostrar los datos, en este artículo comenta que VRML no es lo suficientemente potente para crear el GIS, por

ello se ayuda de la extensión X3D para resolver los problemas con los que se encuentra usando solo VRML.

También se está investigando la posibilidad del 3D en los gis para tratar el crecimiento de las ciudades, por eso en el Institute of 3D Information Acquisition & Application el investigador Wang Yanbing plantea en “3D GIS Spatial Modeling for City Surface and Subsurface Integration” la creación de un GIS 3D dividido en dos ramas, 3D City GIS se encargaría de la parte de la superficie donde con diversas herramientas se mostraría la información de la estructura de la ciudad: calles, edificios, etc. y en la otra rama tenemos 3D Geological Modeling para tratar la parte del subsuelo, en la actualidad las ciudades crecen hacia arriba y hacia abajo y esa parte de la que se encarga esta rama, de las partes subterráneas de la ciudad.

En el paper “3D GIS Database Model for Efficient Management of Large Scale Underground Spatial Data” los autores Gang Liu, Qing Zhu, Zhenwen He, Yeting Zhang, Chonglong Wu, Xiaoming Li y Zhengping Weng explican ya la idea de usar una base de datos para un almacenamiento más eficiente de la información de las capas terrestres separando los datos por capas, materiales, etc.

Por otra parte Heather Richards-Rissetto, Favio Remondino, Giorgio Agugiaro, Jim Robertsson, Jennifer von Schwerin y Gabrio Giardi en su escrito “Kinect and 3D GIS in Archaeology” ya tratan el tema de crear un GIS 3D usando como capturador de modelos la herramienta Kinect de Microsoft, en éste mismo paper, hablan del MayaArch 3D tool, un GIS que han usado en su proyecto sobre el que se hablará más adelante.

En este proyecto, lo que se ha intentado ha sido crear un GIS web usando un motor de videojuegos como Unity 3D para trabajar con la información almacenada en una base de datos.

Iniciativas con motores de Videojuegos

Hasta ahora la mayoría de los GIS han sido en 2D, debido a esto nunca ha sido necesario el manejo de gran cantidad de información gráfica, con el auge en los últimos años y siendo que el mundo es en 3 dimensiones se está investigando en diversas posibilidades para crear GIS que acepten una nueva dimensión, la profundidad.

Como se ha comentado anteriormente en ésta última década están apareciendo multitud de GIS en la vida cotidiana de la gente, el GPS se integró rápidamente en los hogares de la gente y es un claro ejemplo del boom de los sistemas GIS, otro ejemplo es google maps y el lanzamiento de los mapas propios de Apple o los de Microsoft. Con el lanzamiento de los mapas de Apple que incluían una propiedad para poder ver los

edificios en 3D usando lo que ellos llaman Flyover de la que no hay mucha información, se abrió la veda en la expansión de los GIS en 3D.

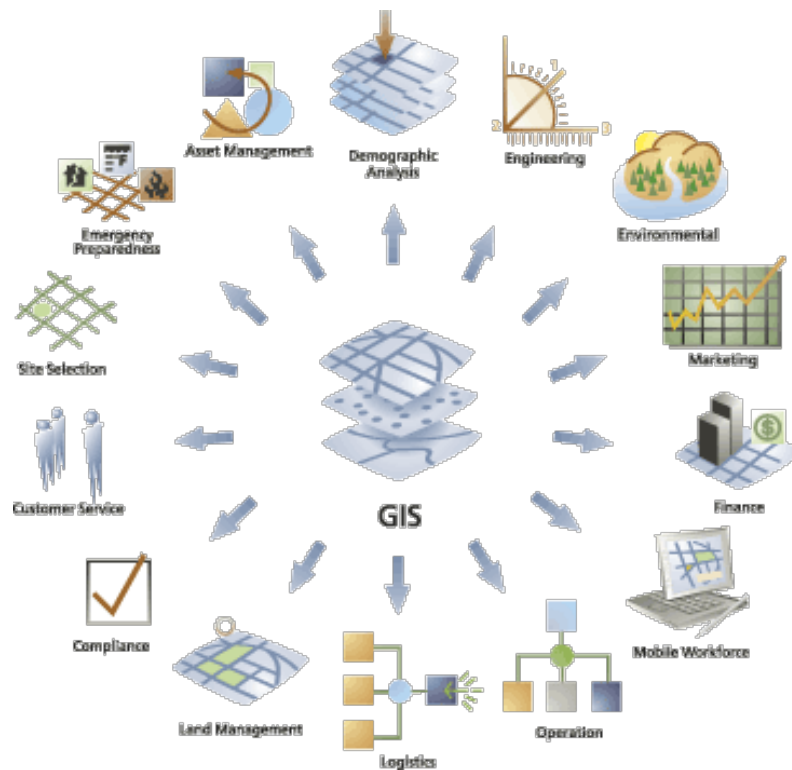


Figura 5: Aplicaciones de un GIS (extraída de <http://gisforyourbusiness.blogspot.com.es>)

Por su parte Google lanzó la beta de MapasGL que es la posibilidad de ver los edificios en 3D por medio de la extrusión de las bases de éstos, que no es más que coger el modelo 2D, en este caso la base del edificio y duplicarlo en capas hacia arriba, dando así lugar a la tercera dimensión, todo esto se ha realizado con la ayuda de WebGL que es un API basado en OpenGL para la web.

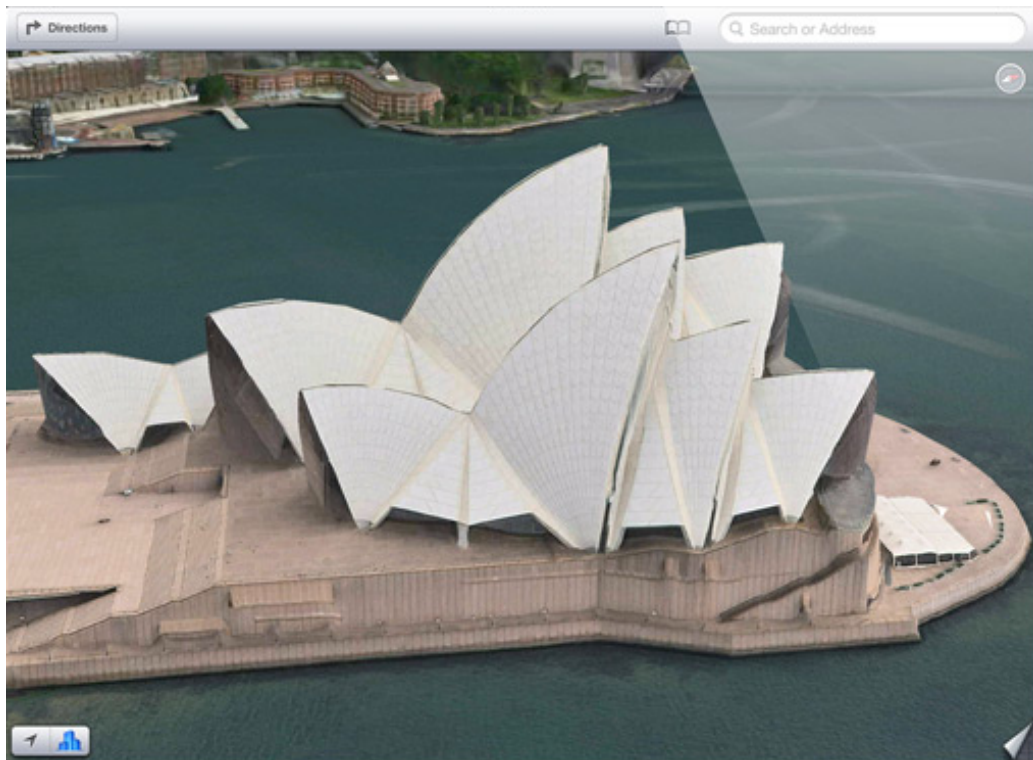


Figura 6: Mapas de Apple (extraída de <http://www.apple.com/es/ios/maps/>)

En el mercado existen ya GIS que usan motores de videojuegos como <http://www.geoweb3d.com> que une un GIS, ArcGIS, otra parte Web donde se guardan los datos y la parte 3D, que es a grandes rasgos lo que se pretende con este proyecto, con la diferencia que el motor usado por nosotros, Unity es mucho más versátil y se puede emplear en varias plataformas, ya que el de geoweb3d es sólo para Windows y necesita mucha potencia 3D.



Figura 7: Estructura de GeoWeb3D (extraída de [http://www.geoweb3d.com/products/#lightbox\[small_gallery\]/4/](http://www.geoweb3d.com/products/#lightbox[small_gallery]/4/))

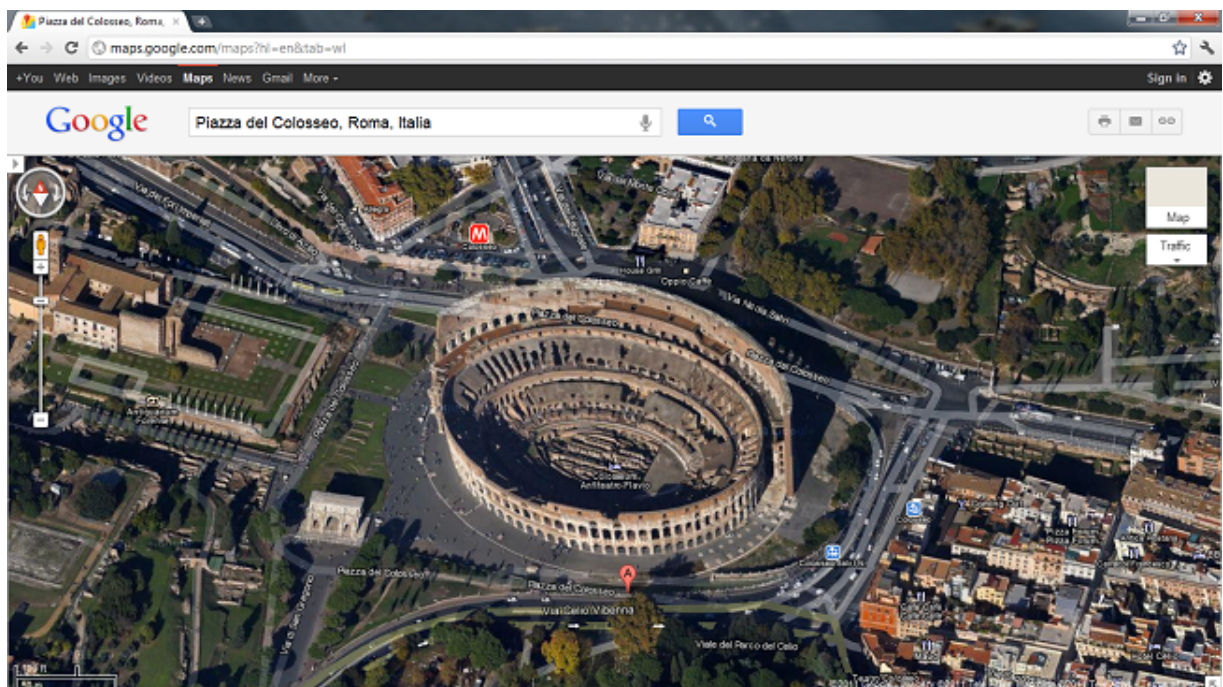


Figura 8: Mapas de Google (captura de pantalla de google maps)

Otra ventaja respecto a otros GIS 3D que ya están en el mercado es que el desarrollado en este proyecto al ser web permite una mayor universalidad que los demás, a parte de poder incluir información extra que no tiene porqué estar relacionada con la geografía, como el estado de conservación de los edificios, material de construcción, etc.

Estas condiciones son las que reúne el GIS MayaArch 3D tool citado anteriormente, este sistema es un proyecto multidisciplinar e internacional que pretende crear una compleja base de datos online para la comunidad de investigadores y público en general. Empezó en 2009 con un arqueólogo y un historiador de la universidad de nuevo México y se han ido uniendo varios colaboradores que han trabajado con ellos para crear la beta que hay ahora en funcionamiento en <http://mayaarch3d.unm.edu/mayaarch3dtool.php> .

Éste proyecto ha sido desarrollado en Unity y los modelos tomados con kinect, las limitaciones de éste dispositivo como capturador 3D hace que los modelos tengan todos un aspecto muy cuadrado y que las texturas tengan muy pocos matices. De momento hay disponibles 2 herramientas, una lupa, para ver información sobre partes de la estructura arquitectónica dividida por tipo, grupos, años, etc. y una regla para medir distancias. Otra posibilidad que brinda este GIS es la de pasear por partes delimitadas del complejo Maya.

Aunque esta herramienta se parece a la aquí desarrollada, la cámara de MayaArch 3D tool es poco precisa, la información muy confusa a la hora de mostrarse y el funcionamiento demasiado complejo y poco intuitivo.



Figura 9: Captura de pantalla de MayaArch 3D

Aplicaciones

Los sistemas GIS, gracias a su versatilidad pueden mostrar información variada, es útil en múltiples ámbitos, algunos de ellos comentados anteriormente. Uno de los más propagados últimamente es el de los GPS, un sistema global de navegación por satélite que permite determinar la posición de un objeto, una persona o un vehículo con una precisión de hasta unos centímetros. Es un hardware de pequeño tamaño con una pantalla que contiene una memoria donde se almacenan los mapas para a continuación, mediante un sistema de satélites y unos protocolos de comunicación se geo localiza el objeto para el que está configurado y luego se sitúa en el mapa del dispositivo.

Las entidades públicas han implementado componentes de los GIS en la construcción y mantenimiento de planos digitales de cartografía. Dichos planos son puestos a disposición de empresas a las que puedan resultar de utilidad estos productos con la condición de que estas entidades se encargan posteriormente y de proveer versiones actualizadas de manera óptima.

Algunos de los primeros sistemas GIS fueron utilizados por las empresas encargadas del desarrollo, mantenimiento y administración de redes de electricidad, gas, agua, teléfono, alcantarillado, etc. En este caso, los sistemas GIS almacenan información alfanumérica de servicios relacionados con las distintas representaciones gráficas de los mismos. Estos sistemas almacenan información relativa a la conectividad de los elementos representados gráficamente, con el fin de realizar un análisis de redes. La elaboración de mapas, así como la posibilidad de realizar una consulta combinada de información son las funciones más comunes para estos sistemas, también son utilizados en trabajos de ingeniería, inventarios, planificación de redes, gestión de mantenimiento, etc.

Los GIS son aplicaciones dirigidas a la gestión de entidades territoriales y permiten un rápido acceso a la información gráfica y alfanumérica, y suministran herramientas para el análisis espacial de la información. Facilitan labores de mantenimiento de infraestructuras, mobiliario urbano, etc. y permiten realizar una optimización en los trabajos de mantenimiento de empresas de servicios.

También son aplicaciones implementadas por instituciones de medio ambiente, que facilitan la evaluación del impacto ambiental en la ejecución de proyectos. Integrados con sistemas de adquisición de datos permiten el análisis en tiempo real de contaminantes, a fin de tomar las precauciones y medidas del caso. Facilitan una ayuda fundamental en trabajos tales como reforestación, explotaciones agrícolas, estudios de representatividad, caracterización de ecosistemas, estudios de fragmentación, estudios de especies, etc.

Otros GIS se implementan dirigidos a la gestión de servicios de impacto social, tales como servicios sanitarios, centros escolares, hospitales, centros deportivos, culturales, lugares de concentración en casos de emergencias, centros de recreo, entre otros y suministran información sobre las sedes ya existentes en una determinada zona y ayudan a la planificación de la localización de nuevos centros. Un buen diseño de estos GIS aumentan la productividad al optimizar recursos, ya que permiten asignar de forma adecuada y precisa los centros de atención a usuarios cubriendo de forma eficiente la totalidad de la zona de influencia.

Estos programas se usan también para la explotación minera, el diseño de éstos facilitan el manejo de un gran volumen de información generada en varios años de explotación intensiva de un banco minero, suministrando funciones para la realización de análisis de elementos puntuales (sondeos o puntos topográficos), lineales (perfiles, tendido de electricidad), superficies (áreas de explotación) y volúmenes (capas geológicas).

Hay GIS utilizados para modelar la conducta del tráfico determinando patrones de circulación por una vía en función de las condiciones del tráfico y longitud. Asignando un costo a los puntos en los que puede existir un semáforo, se puede obtener información muy útil relacionada con análisis de redes.

Para estudios demográficos también se pueden usar los sistemas GIS teniendo un conjunto diverso de aplicaciones cuyo vínculo es la utilización de las variadas características demográficas, y en concreto la distribución espacial, para la toma de decisiones. Algunas de estas aplicaciones pueden ser: el análisis para la implantación de negocios o servicios públicos, zonificación electoral, etc. El origen de los datos regularmente corresponde a los censos poblacionales elaborados por alguna entidad gubernamental.

La base de datos de los clientes potenciales de determinado producto o servicio relacionado con la información geográfica resulta indispensable para planificar una adecuada campaña de marketing o el envío de correos promocionales, se podrían diseñar rutas óptimas a seguir por comerciales, anuncios, publicidad, etc.

Los bancos son buenos usuarios de los GIS debido a que requieren ubicar a sus clientes y planificar tanto sus campañas como apertura de nuevas sucursales incluyendo información sobre las sucursales de la competencia.

La planimetría tiene como objetivo la representación bidimensional del terreno proporcionándole al usuario la posibilidad de proyectar su trabajo sobre un papel o en pantalla sin haber estado antes en el sitio físico del proyecto. El fin de la planimetría es que el usuario tenga un fácil acceso a la información, por ejemplo, saber qué cantidad de terrenos desocupados se encuentran en el lugar, o qué cantidad de postes telefónicos se necesitan para ampliar su red, o qué cantidad de cable necesita para

llegar hasta un cliente, o emplearlo en soluciones móviles, o utilizarlo como plataforma de archivos GIS. En otras palabras, permite al usuario visualizar de forma clara y con gran exactitud la información que se encuentra dentro de su proyecto. Existen distintos tipos de planimetría, que van desde la más básica a la más completa. La elección del tipo de planimetría depende del tipo de información que el usuario vaya a necesitar en su proyecto.

En España hay varios proyectos relacionados con los GIS, uno de ellos es el SIGNA, perteneciente al Plan Estratégico del IGN-CNIG desde el año 2005 que tiene como objetivo integrar las bases de datos geográficos digitales del IGN en un GIS, especialmente las Bases Cartográficas y Topográficas, con la finalidad de utilizar sus datos de forma conjunta y coherente, y dotarles de topología y continuidad más allá de la unidad mínima de producción/actualización.

Como consecuencia, se dispone de un GIS básico, integrado y coherente para la consulta y actualización de los datos accesibles desde las unidades pertinentes del IGN-CNIG y para la consulta de usuarios externos mediante una aplicación accesible a través de internet.

El proyecto SIGNA (Sistema de Información Geográfica Nacional) consta de dos grandes áreas de trabajo que a grandes rasgos podrían definirse como: la parte encargada de la continuidad y estructuración de los datos, denominada "estructuración"; y la parte de desarrollo y mantenimiento de las aplicaciones y clientes que permiten al público acceder y analizar los datos geográficos.

Otro de los proyectos GIS a nivel nacional es el SINAIE, el Sistema de Información del Atlas Nacional de España, es un proyecto de innovación y desarrollo que da soporte tecnológico a la producción y publicación del Atlas Nacional de España (ANE), así como a otros productos y servicios ofrecidos por el Área de Cartografía Temática y Atlas Nacional del IGN.

A través del portal SIANEWEB el usuario puede acceder por medio de consultas personalizadas a la información geográfica actualizada y a los contenidos digitales de forma individual (un mapa, una imagen, un epígrafe, etc.), ya que se presenta en forma de catálogo; o crear su propia selección de contenidos, sin necesidad de esperar a que un determinado grupo temático esté completamente actualizado.

Los sistemas GIS son usados también en los estudios del medio ambiente y los recursos naturales como en aplicaciones forestales, un caso específico es el control de plagas, en este caso se utiliza principalmente para la determinación de zonas de riesgo, el análisis de comportamiento y evolución de plagas. También se pueden diseñar redes de trampas y puntos de control.

Tiene una importancia especial en este caso, por la información que facilitan, la fotografía aérea y las imágenes de satélite.

También se usan para la estimación de la biomasa residual ya que ésta se puede aprovechar energéticamente, presentando así un doble beneficio, en primer lugar si eliminamos estos residuos de los bosques podrán minimizarse de los impactos ambientales derivados de plagas o incendios y en segundo lugar, el propio aprovechamiento de esta madera como fuente de energía renovable, con las ventajas que esto conlleva, como el balance neutro de CO₂.

Los GIS en ambientes forestales también se utilizan para localizar y estimar el grado de deforestación y sus efectos en los ecosistemas forestales, principalmente en aquellos más aislados, pudiéndose simular situaciones de riesgo para ver como evolucionarían hipotéticamente, y poder estimar sus consecuencias, para poder determinar medidas de restauración.

Los incendios forestales producidos por actividades humanas, y causas naturales, van además incrementándose año tras año por los efectos del cambio climático. Los GIS se generan en este campo para prevenir, controlar y gestionar estos incendios, utilizándose como herramienta de apoyo a la toma de decisiones.

Los estudios sobre incendios forestales surgieron en los años 20 en EEUU y Canadá. Al principio se utilizaban para crear índices de peligro basados en datos meteorológicos. Estos sistemas han ido evolucionando hasta convertirse en sistemas de riesgo que además de utilizar datos meteorológicos, utilizan otros factores como el comportamiento del fuego, topografía, actividad humana, modelos de combustible, etc. Así como la información obtenida de las imágenes de satélite.

Algunos programas de este tipo de gran importancia son:

EEUU: NFDRS (National Fire Danger Rating System).

Canadá: CFFDRS (Canadian Forest Fire Danger Rating System).

Europa: EFFIS (European Forest FIRE Information System).

Otro uso de los sistemas GIS es el de control de cambio de uso del suelo, inventarios de uso, planificación de explotaciones agrícolas, un ejemplo de esto sería el SIGPAC que es un GIS de parcelas agrícolas, que permite identificar geográficamente las parcelas declaradas por los agricultores y ganaderos, en cualquier régimen de ayudas relacionado con la superficie cultivada o aprovechada por el ganado. Concebido inicialmente con el propósito de facilitar a los agricultores la presentación de solicitudes, con soporte gráfico, así como para facilitar los controles administrativos y

sobre el terreno, el SIGPAC se ha convertido en una herramienta de enorme utilidad en campos diferentes del agrario (geología, infraestructuras, urbanismo...), lo que obedece a su concepción y desarrollo, en el que se hace uso continuo y permanente de las tecnologías más avanzadas en información geográfica automatizada.

Tenemos también el SIG OLEÍCOLA que es el Sistema de Información Geográfica Oleícola Español permite la consulta de las parcelas catastrales oleícolas tras escoger la provincia, el municipio y el polígono catastral deseado. Actualmente, la información contenida en el SIG Oleícola se ha integrado en el SIGPAC.

Por otro lado tenemos el SIGMAPA, el objetivo principal del Sistema de Información Geográfica del Área del Medio Rural y Marino (SIGMAPA), es la publicación y difusión de la información geográfica relativa a este ámbito del Ministerio de Medio Ambiente y Medio Rural y Marino en Internet. La información que se publica es muy variada y atiende a distintos tipos de disciplinas: regadíos, plan de choque, mapas de cultivos, agricultura, datos agroclimáticos, denominaciones de origen, industrias agroalimentarias, gestión de ayudas, cuencas hidrográficas, cartografía general, etc.

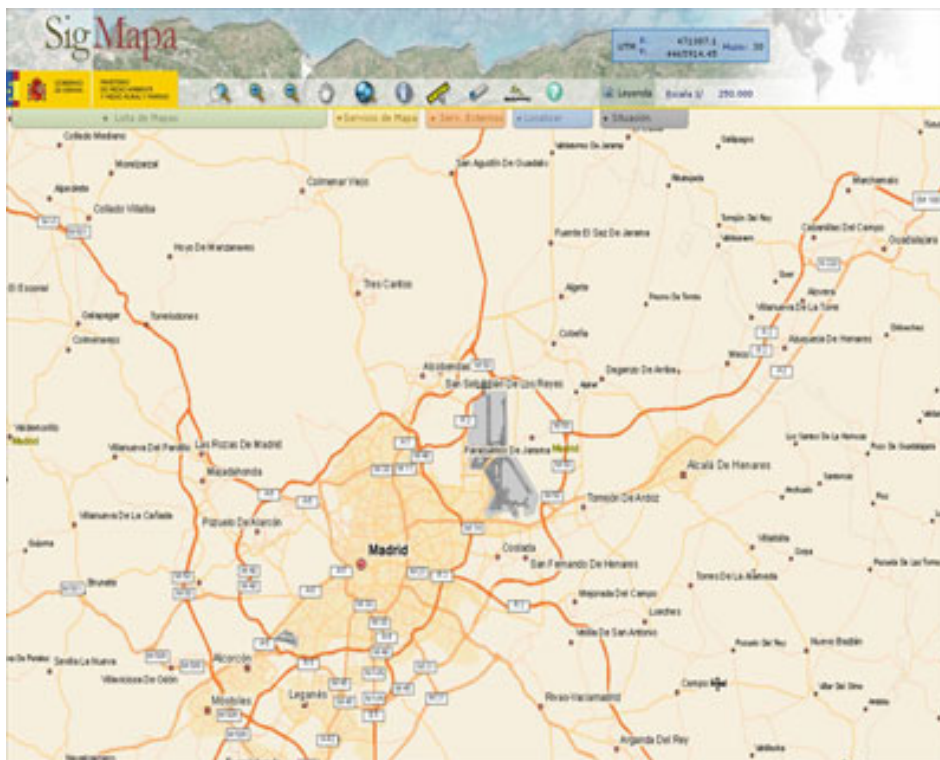


Figura 10: captura de SigMapa

El SIGMAPA integra la información geográfica del medio agrario, pesquero y alimentario accesible mediante un único visor desde Internet, favoreciendo además la

interoperabilidad con otros organismos públicos (Instituto Geográfico Nacional, Catastro, Instituto Tecnológico Geominero, etc.). Incorpora funcionalidad avanzada para fines de promoción agroalimentaria.

La aplicación SIGA, Sistema de Información Geográfico Agrario, desarrollada a iniciativa de la Subdirección General de Cultivos Herbáceos e Industriales adscrita a la Dirección General de Recursos Agrícolas y Ganaderos del MARM ofrece información cartográfica y alfanumérica, mapas temáticos sobre variables agroclimáticas, informas sobre municipios y estaciones meteorológicas, etc.

Los vertederos, plantas de tratamiento biológico, o plantas de tratamiento físico-químico de residuos, entre otros, pueden establecerse también mediante GIS.

En este caso deben tenerse en cuenta puntos que produzcan grandes cantidades de residuos, como pueden ser industrias y plantas de depuración de aguas residuales. También habrá que tener en cuenta vías de comunicación, instalaciones eléctricas de alta tensión (para la utilización de energía), otras plantas ya existentes, la cercanía de núcleos urbanos, y cauces o balsas de agua subterránea, flora y fauna y la visibilidad de la instalación, para calcular también el impacto medioambiental en la zona.

Otra posible aplicación de los GIS es la determinación de la concentración de los diferentes contaminantes atmosféricos de una zona, así como las fuentes que los producen: de los que se derivan para poder realizar métodos de reducción de la misma, como puede ser el uso de carbones activados, condensadores, recirculación de gases de escape, etc.

Los GIS son muy útiles en los estudios de zonificación de especies, es decir, en los que tenemos que separar especies que están muy próximas por zonas diferenciadas.

Si las especies son vegetales, se aplican distintas variables, como la altura, pendiente, amplitud del relieve, etc. Conocer la distribución de las especies forestales, es un factor muy importante a tener en cuenta a la hora del manejo forestal.

En el caso de especies animales, se pueden utilizar para realizar la modelización de un tipo de hábitat potencial de una especie en concreto. Su importancia radica en poder estudiar, disponiendo de la mayor información posible, especies protegidas o en peligro de extinción, teniendo en cuenta, por ejemplo, zonas de cría, de hibernación y de alimentación.

Mediante los GIS se puede determinar el impacto que puede suponer una obra determinada sobre el medio ambiente. Se puede determinar, por ejemplo, el impacto que causaría el trazado de una autovía sobre un paraje.

Esto se haría teniendo en cuenta, capas sucesivas de los diferentes factores que interactúan, como pudieran ser: riesgo sísmico, riesgo de deslizamiento del suelo,

riesgo geológico, riesgo de inundación, flora y fauna singular, municipios y cauces de ríos, así como la red actual de carreteras existentes. Con la superposición de estas capas y el estudio de los datos asociados a ellas, se podrían establecer las distintas zonas del impacto producido.

Aplicación Desarrollada

Propuesta de solución

Lo que se busca con este proyecto es abarcar un rango en los usos de los GIS poco explorado, de la cantidad de ejemplos comentados con anterioridad, hay pocos ejemplos de GIS empleados para estudios arqueológicos, una de las razones es porque en 2 dimensiones la información arqueológica suministrada por un GIS podrían ser las coordenadas donde se encuentra la ruina arqueológica, posible estructura del emplazamiento sobre el mapa, etc.

Sería mucho más interesante para los arqueólogos si se pudiera crear un modelo 3D del edificio a estudiar con la posibilidad de añadir estructuras posibles sobre éste en caso de que no se encuentre entero. También sería de gran ayuda poder relacionar información con la estructura, para así conseguir a parte de tener la información relativa a la localización e información del terreno donde se encuentra, tener en la misma aplicación la información relativa a la composición de la estructura, detalles urbanísticos, edades de las construcciones, etc.

Por todo ello ha nacido este proyecto, su finalidad es ayudar a los arqueólogos en la tarea de administrar la información adquirida de estudiar los yacimientos arqueológicos y sus entornos, dando la posibilidad de adjuntar la información tanto a los mapas como a los modelos en 3D adquiridos después de escanear los restos arquitectónicos con un escáner 3D.

Después de pensar la mejor distribución para la información se ha llegado a la conclusión de que un sistema cliente–servidor sería la mejor organización para la idea a desarrollar, esto conlleva que se divida el proyecto en 3 partes, por una parte se desarrollará la base de datos, parte primordial del proyecto, ésta base de datos estará dividida en varias tablas organizadas para la mejor distribución de la información y una catalogación óptima de los datos, la base de datos se ubicará en un servidor remoto con la finalidad de poder acceder a ella desde cualquier lugar y dispositivo, ya que con el avance de la tecnología hoy en día no es difícil encontrar numerosos dispositivos con acceso a la red.

Por otro lado tendremos el Back–End, se trata de una aplicación desarrollada en Visual Studio para la introducción de la información en la base de datos. Todo el proyecto está orientado a arqueólogos y arquitecto, por ello se pretende facilitarles el trabajo en lugar de dificultárselo, es por ello que esta parte pretende hacer simple la tarea de introducir la información en la base de datos. Ésta aplicación dispondrá de una interfaz intuitiva y de uso sencillo pero completa, se dispondrá de una parte de gestión de usuarios para controlar la modificación e introducción de datos.

La otra parte de la aplicación será un visualizador de la información, para permitir un mejor acceso a los datos por ello, se ha pensado en crear una web conectada con la base de datos usando PHP, en ésta web se dispondrá de una parte para navegar por la información y otra parte para poder visualizar los modelos. Como en esta herramienta los modelos en 3D son muy importantes y la información hará referencia a ellos, mediante el motor de videojuegos Unity 3D se desarrollará un visualizador de modelos, donde se dispondrá de una vista general del yacimiento a estudiar y la posibilidad de navegar por él y acceder a la información relacionada a cada parte del modelo.

Herramientas utilizadas

Unity

Unity es un motor de videojuego multiplataforma creado por Unity Technologies. Unity está disponible como plataforma de desarrollo para Windows y OS X, y permite crear juegos para Windows, OS X, Linux, Xbox 360, PlayStation 3, Wii, Wii U, iPad, iPhone y Android. Gracias al Plug-In Web de Unity, también se pueden desarrollar juegos de navegador, para Windows y Mac. Su última versión, la 4.2, fue lanzada en julio de 2013. Desde la página oficial se pueden descargar dos versiones: Unity y Unity Pro.

Unity Technologies fue fundada en 2004 por David Helgason (CEO), Nicholas Francis (CCO), y Joachim Ante (CTO) en Copenhague, Dinamarca después de su primer juego, GooBall, que no tuvo éxito. Los tres reconocieron el valor del motor y las herramientas de desarrollo y se dispuso a crear un motor que todos pudiéramos usar a un precio asequible. Unity Technologies ha recibido financiación de la talla de Sequoia Capital, Capital WestSummit y Socios iGlobe.

El éxito de Unity ha llegado en parte debido al enfoque en las necesidades de los desarrolladores independientes que no pueden crear ni su propio motor del juego ni las herramientas necesarias o adquirir licencias para utilizar plenamente las opciones que aparecen disponibles. El enfoque de la compañía es “democratizar el desarrollo de juegos”, y hacer el desarrollo de contenidos interactivos en 2D y 3D lo más accesible posible a tantas personas en todo el mundo como sea posible.

En 2008, con auge del iPhone, Unity fue uno de los primeros desarrolladores de motores en empezar a apoyar a la plataforma en su totalidad. Unity está siendo utilizado por el 53'1% de los desarrolladores (según la encuesta de Game Developer tecnología móvil y social), con cientos de juegos lanzados en dispositivos Android y iOS.

En 2009, Unity comenzó a ofrecer una versión de su producto de forma gratuita. El número de desarrolladores registrados empezó a crecer rápidamente tras el anuncio. En abril de 2012, Unity se dio cuenta del nuevo nivel de popularidad a medida que el recuento de los desarrolladores registrados alcanzaron 1 millón, 300.000 de los cuales utilizan Unity sobre una base mensual regular.

La versión de Unity se lanzó en la Conferencia Mundial de Desarrolladores de Apple en 2005. Fue construido solamente para funcionar y generar proyectos en los equipos de la plataforma Mac y obtuvo el éxito suficiente como para continuar con el desarrollo del motor y herramientas. Unity 3 fue lanzado en septiembre de 2010 y se centró en empezar a introducir más herramientas que los estudios de alta gama por lo general

tienen a su disposición, con el fin de captar el interés de los desarrolladores de más grandes, mientras que proporciona herramientas para equipos independientes y más pequeñas que normalmente serían difíciles de conseguir en un paquete asequible. La última versión de Unity, Unity 4, lanzada a finales de 2012, se anunció en junio de 2012 e incluye añadidos como Mecanim animation, soporte para DirectX 11 y soporte para juegos en Linux.

Unity soporta integración con 3ds Max, Maya, Softimage, Blender, Modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks y Allegrithmic Substance. Los cambios realizados a los objetos creados con estos productos se actualizan automáticamente en todas las instancias de ese objeto durante todo el proyecto sin necesidad de volver a importar manualmente.

El motor gráfico utiliza Direct3D (Windows), OpenGL (Mac, Linux), OpenGL ES (Android, iOS), y APIs propietarias (Wii). Tiene soporte para mapeado de relieve, reflexión de mapeado, mapeado por paralaje, pantalla de espacio oclusión ambiental (SSAO), sombras dinámicas utilizando mapas de sombras, render a textura y efectos de post-procesamiento de pantalla completa.

Se usa ShaderLab language para el uso de shaders, soporta tanto programación declarativa de los programas de función fija de tuberías y shader GLSL o escritas en Cg. Un shader puede incluir múltiples variantes y una especificación declarativa de reserva, lo que permite a Unity detectar la mejor variante para la tarjeta de vídeo actual y sino son compatibles, recurrir a un shader alternativo que puede sacrificar características para una mayor compatibilidad.

El soporte integrado para Nvidia (antes Ageia), el motor de física PhysX, (a partir de Unity 3.0) con soporte en tiempo real para mallas arbitrarias y sin piel, ray casts gruesos, y las capas de colisión.

El scripting viene a través de Mono. El script se basa en Mono, la implementación de código abierto de .NET Framework. Los programadores pueden utilizar UnityScript (un lenguaje personalizado inspirado en la sintaxis ECMAScript), C# o Boo (que tiene una sintaxis inspirada en Python). A partir de la versión 3.0 añade una versión personalizada de MonoDevelop para la depuración de script.

Unity también incluye Unity Asset Server – una solución de control de versiones para todos los assets de juego y scripts, utilizando PostgreSQL como backend, un sistema de audio construido con la biblioteca FMOD, con capacidad para reproducir audio comprimido Ogg Vorbis, reproducción de vídeo con códec Theora, un motor de terreno y vegetación, con árboles con soporte de billboarding, determinación de cara oculta con Umbra, una función de iluminación lightmapping y global con Beast, redes multi jugador RakNet y una función de búsqueda de caminos en mallas de navegación.

Unity 3.5 fue uno de los grandes lanzamientos que ha salido de la plataforma de desarrollo de Unity y añade un montón de nuevas características y mejoras en la tecnología existente. Estos incluyen el sistema de partículas Shuriken, navmesh para pathfinding y evasión de obstáculos, iluminación del espacio lineal (gamma correcta), renderizado HDR, multi-threaded rendering, sondas de luz, despliegue Google Native Client, reescrito desecho oclusión, una función de nivel de detalle apoyo, Adobe Flash Player add-on para vista previa, perfilador de GPU, y lightmaps direccionales.

Unity 4 fue anunciado el 18 de junio de 2012 e incluye varias características adicionales a la tecnología de Unity. El proceso de liberación de Unity 4, al igual que las versiones anteriores. Se incluye con varias actualizaciones con las características adicionales a lo largo de su vida útil, como la nueva interfaz gráfica de usuario retenido, lo cual es debido en una futura actualización 4.x.

Mecanim es la tecnología de animación de Unity, que ha estado en desarrollo durante años, primero por la empresa del mismo nombre, a continuación, por las oficinas de Unity en Canadá, después de su adquisición. La tecnología está diseñada para llevar el movimiento fluido y natural de los personajes con una interfaz eficiente. Mecanim incluye herramientas para la creación de máquinas de estados, árboles de mezcla, manipulación de los conocimientos indígenas y retargeting automático de animaciones, desde el editor de Unity.

Además, una serie de animaciones redestinables estarán disponible en el Unity Asset Store de Unity tras el lanzamiento de la herramienta. Muchos de estos archivos de animación de captura de movimiento son proporcionados sin costo por Unity Technologies. Otros proveedores del Asset Store también ofrecerá animaciones para su uso con Mecanim, ya sea de forma gratuita o de pago.

Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J# y Visual Basic .NET, al igual que entornos de desarrollo web como ASP .NET aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

A partir de la versión 2005 Microsoft ofrece gratuitamente las Express Editions, que son varias ediciones básicas separadas por lenguajes de programación o plataforma enfocadas para novatos y entusiastas. Estas ediciones son iguales al entorno de desarrollo comercial pero sin características avanzadas. Adicionalmente, Microsoft ha puesto gratuitamente a disposición de todo el mundo una versión reducida de MS SQL Server llamada SQL Server Express Edition cuyas principales limitaciones son que no soporta bases de datos superiores a 4 GB de tamaño, únicamente se ejecuta en un procesador y emplea 1 GB de RAM como máximo, y no cuenta con el Agente de SQL Server.

Visual Studio 6.0 se lanzó en 1998 y fue la última versión en ejecutarse en la plataforma Win9x. Los números de versión de todas las partes constituyentes pasaron a 6.0, incluyendo Visual J++ y Visual InterDev, que se encontraban en las versiones 1.1 y 1.0 respectivamente. Esta versión fue la base para el sistema de desarrollo de Microsoft para los siguientes 4 años, en los que Microsoft migró su estrategia de desarrollo al .NET Framework.

Visual Studio 6.0 fue la última versión en que Visual Basic se incluía de la forma en que se conocía hasta entonces; versiones posteriores incorporarían una versión muy diferente del lenguaje con muchas mejoras, fruto de la plataforma .NET. También supuso la última versión en incluir Visual J++, que proporcionaba extensiones de la plataforma Java, lo que lo hacía incompatible con la versión de Sun Microsystems. Esto acarreó problemas legales a Microsoft, y se llegó a un acuerdo en el que Microsoft dejaba de comercializar herramientas de programación que utilizaran la máquina virtual de Java.

Aunque el objetivo a largo plazo de Microsoft era unificar todas las herramientas en un único entorno, esta versión añadía un entorno más a Visual Studio 5.0: Visual J+ y Visual Interdev se separaban del entorno de Visual C++, al tiempo que Visual FoxPro y Visual Basic seguían manteniendo su entorno específico.

En la versión Visual Studio .NET (2002) se produjo un cambio sustancial, puesto que supuso la introducción de la plataforma .NET de Microsoft. .NET es una plataforma de ejecución intermedia multilenguaje, de forma que los programas desarrollados en .NET no se compilan en lenguaje máquina, sino en lenguaje intermedio (CIL – Common Intermediate Language) denominado Microsoft Intermediate Language (MSIL). En una aplicación MSIL, el código no se convierte a lenguaje máquina hasta que ésta se ejecuta, de manera que el código puede ser independiente de plataforma (al menos de las soportadas actualmente por .NET). Las plataformas han de tener una implementación de Infraestructura de Lenguaje Común (CLI) para poder ejecutar programas MSIL. Actualmente se pueden ejecutar programas MSIL en Linux y Mac OS X usando implementaciones de .NET que no son de Microsoft, tales como Mono y DotGNU.

Visual Studio .NET 2002 supuso también la introducción del lenguaje C#, un lenguaje nuevo diseñado específicamente para la plataforma .NET, basado en C++ y Java. Se presentó también el lenguaje J# (sucesor de J++), el cual, en lugar de ejecutarse en una máquina virtual de Java, se ejecuta únicamente en el framework .NET. El lenguaje Visual Basic fue remodelado completamente y evolucionó para adaptarse a las nuevas características de la plataforma .NET, haciéndolo mucho más versátil y dotándolo con muchas características de las que carecía. Algo similar se llevó a cabo con C++, añadiendo extensiones al lenguaje llamadas Managed Extensions for C++ con el fin de que los programadores pudieran crear programas en .NET. Por otra parte, Visual FoxPro pasa a comercializarse por separado.

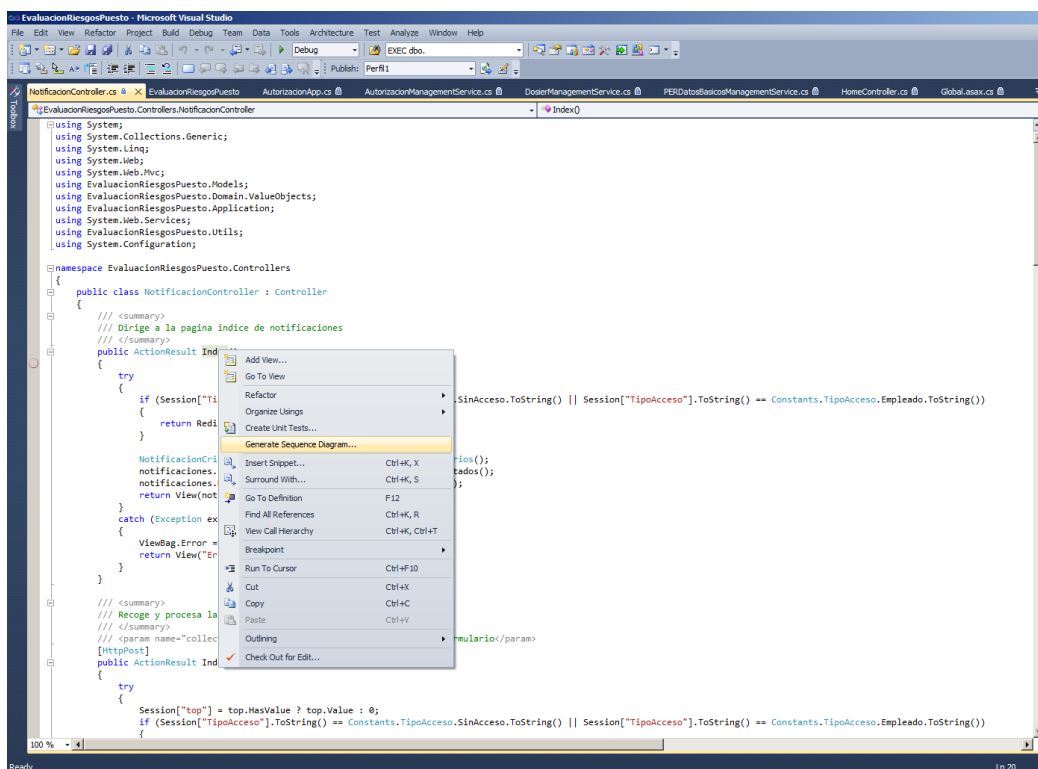


Figura 11: Entorno de programación de Visual Studio

Todos los lenguajes se unifican en un único entorno. La interfaz se mejora notablemente en esta versión, siendo más limpia y personalizable.

Visual Studio .NET 2003 supone una actualización menor de Visual Studio .NET. Se actualiza el .NET Framework a la versión 1.1, también se añade soporte con el fin de escribir aplicaciones para determinados dispositivos móviles, ya sean con ASP .NET o con el .NET Compact Framework. Además, el compilador de Visual C++ se mejora para cumplir con más estándares: el Visual C++ Toolkit 2003.

Visual Studio 2003 se lanza en 4 ediciones: Academic, Professional, Enterprise Developer y Enterprise Architect. La edición Enterprise Architect incluía una implementación de la tecnología de modelado Microsoft Visio, que se centraba en la creación de representaciones visuales de la arquitectura de la aplicación basadas en UML. También se introdujo “Enterprise Templates”, para ayudar a grandes equipos de trabajo a estandarizar estilos de programación e impulsar políticas de uso de componentes y asignación de propiedades.

Visual Studio 2005 se empezó a comercializar a través de internet a partir del 4 de octubre de 2005 y llegó a los comercios a finales del mes de octubre en inglés. En castellano no salió hasta el 4 de febrero de 2006. Microsoft eliminó la coletilla .NET de su nombre, pero eso no indica que se alejara de la plataforma .NET, de la cual se incluyó la versión 2.0.

La actualización más importante que recibieron los lenguajes de programación fue la inclusión de tipos genéricos, similares en muchos aspectos a las plantillas de C++. Con esto se consigue encontrar muchos más errores en la compilación en vez de en tiempo de ejecución, incitando a usar comprobaciones estrictas en áreas donde antes no era posible. C++ tiene una actualización similar con la adición de C++/ CLI como sustituto de C# manejado.

Se incluye un diseñador de implantación, que permite que el diseño de la aplicación sea validado antes de su implantación. También se incluye un entorno para publicación web y pruebas de carga para comprobar el rendimiento de los programas bajo varias condiciones de carga.

Visual Studio 2005 también añade soporte para arquitecturas de 64 bits. Aunque el entorno de desarrollo sigue siendo una aplicación de 32 bits, Visual C++ 2005 soporta compilación para x86-64 (AMD64, Intel 64) e IA-64 (Itanium). El SDK incluye compiladores de 64 bits así como versiones de 64 bits de las librerías.

Visual Studio 2005 tiene varias ediciones radicalmente distintas entre sí; Express, Standard, Professional, Tools for Office y 5 ediciones Visual Studio Team System. Éstas últimas se proporcionaban conjuntamente con suscripciones a MSDN cubriendo los 4 principales roles de la programación: Architects, Software Developers, Testers y Database Professionals. La funcionalidad combinada de las 4 ediciones Team System se ofrecía como la edición Team Suite. Por otra parte, Tools of the Microsoft Office System está diseñada para extender la funcionalidad Microsoft Office.

Las ediciones Express se han diseñado para principiantes, aficionados y pequeños negocios, todas disponibles gratuitamente a través de la página de Microsoft. Se incluye una edición independiente para cada lenguaje: Visual Basic, Visual C++, Visual C#, Visual J# para programación .NET en Windows y Visual Web Developer para la

creación de sitios web ASP .NET. Las ediciones Express carecen de algunas herramientas avanzadas de programación así como de opciones de extensibilidad.

Visual Studio 2008 fue publicado (RTM) el 17 de noviembre de 2007 en inglés, mientras que la versión en castellano no fue publicada hasta el 2 de febrero de 2008.

El nuevo framework (.NET 3.5) está diseñado para aprovechar las ventajas que ofrece el nuevo sistema operativo Windows Vista a través de sus subsistemas Windows Communication Foundation (WCF) y Windows Presentation Foundation (WPF), el primero tiene como objetivo la construcción de aplicaciones orientadas a servicios, mientras que el último apunta a la creación de interfaces de usuario más dinámicas que las conocidas hasta el momento.

A las mejoras de desempeño, escalabilidad y seguridad con respecto a la versión anterior, se agregan, entre otras, las siguientes novedades:

- La mejora en las capacidades de pruebas unitarias permiten ejecutarlas más rápido independientemente de si lo hacen en el entorno IDE o desde la línea de comandos. Se incluye además un nuevo soporte para diagnosticar y optimizar el sistema a través de las herramientas de pruebas de Visual Studio. Con ellas se podrán ejecutar perfiles durante las pruebas para que ejecuten cargas, prueben procedimientos contra un sistema y registren su comportamiento, y utilizar herramientas integradas para depurar y optimizar.
- Con Visual Studio Tools for Office (VSTO) integrado con Visual Studio 2008 es posible desarrollar rápidamente aplicaciones de alta calidad basadas en la interfaz de usuario (UI) de Office que personalicen la experiencia del usuario y mejoren su productividad en el uso de Word, Excel, PowerPoint, Outlook, Visio, InfoPath y Project. Una completa compatibilidad para implementación con ClickOnce garantiza el entorno ideal para una fácil instalación y mantenimiento de las soluciones Office.
- Visual Studio 2008 permite incorporar características del nuevo Windows Presentation Foundation sin dificultad tanto en los formularios de Windows existentes como en los nuevos. Ahora es posible actualizar el estilo visual de las aplicaciones al de Windows Vista debido a las mejoras en Microsoft Foundation Class Library (MFC) y Visual C++. Visual Studio 2008 permite mejorar la interoperabilidad entre código nativo y código manejado por .NET. Esta integración más profunda simplificará el trabajo de diseño y codificación.
- LINQ (Language Integrated Query) es un nuevo conjunto de herramientas diseñado para reducir la complejidad del acceso a bases de datos a través de extensiones para C++ y Visual Basic, así como para Microsoft .NET Framework. Permite filtrar, enumerar, y crear proyecciones de muchos tipos y colecciones de datos utilizando todos la misma sintaxis, prescindiendo del uso de lenguajes especializados.

- Visual Studio 2008 ahora permite la creación de soluciones multiplataforma adaptadas para funcionar con las diferentes versiones de .NET Framework 2.0 (incluido con Visual Studio 2005), 3.0 (incluido en Windows Vista) y 3.5 (Incluid con Visual Studio 2008).
- .NET Framework 3.5 incluye la biblioteca ASP .NET AJAX para desarrollar aplicaciones web más eficientes, interactivas y altamente personalizadas que funcionen para todos los navegadores más populares y utilicen las últimas tecnologías y herramientas Web, incluyendo Silverlight y Popfly.

Visual Studio 2010 es la versión fue lanzada el 12 de abril de 2010 acompañada por .NET Framework 4.0.

Hasta ahora, uno de los mayores logros de la versión 2010 de Visual Studio ha sido el de incluir las herramientas para desarrollo de aplicaciones para Windows 7, tales como herramientas para el desarrollo de las características de Windows 7 (System.Windows.Shell) y la Ribbon Preview para WPF.

Entre sus más destacables características, se encuentran la capacidad para utilizar múltiples monitores, así como la posibilidad de desacoplar las ventanas de su sitio original y acoplarlas en otros sitios de la interfaz de trabajo.

Además ofrece la posibilidad de crear aplicaciones para muchas plataformas de Microsoft, como Windows, Azure, Windows Phone 7 o SharePoint. Microsoft ha sido sensible a la nueva tendencia de las pantallas táctiles y con este Visual Studio 2010 también es posible desarrollar aplicativos para pantallas multitáctiles.

Entre las ediciones disponibles de Visual Studio 2010 que podemos adquirir se encuentran la versión Ultimate, Premium, Professional, Team Foundation Server 2010, Test Professional 2010 y Team Explorer Everywhere 2010.

phpMyAdmin

phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente se puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 69 idiomas, además se encuentra bajo licencia GPL.

Este proyecto se encuentra vigente desde el año 1998, siendo el mejor evaluado en la comunidad de descargas de SourceForge.net como la descarga del mes de diciembre de 2002. Como esta herramienta corre en máquinas con Servidores Webs y soporte de PHP y MySQL, la tecnología utilizada ha ido variando durante su desarrollo.

Tobias Ratschiller, por entonces consultor de IT y después fundador de Maguma, una compañía de software, comenzó a trabajar en la elaboración de una red administrativa basada en PHP cliente-servidor en MySQL en 1998 e inspirado por Peter Kuppelwieser y su MySQL-Webadmin. Cuando Ratschiller dejó el proyecto por falta de tiempo, así como el phpAdsNew del cual es también su inventor, el phpMyAdmin se había convertido en una de las aplicaciones PHP más populares, y las herramientas de administración MySQL constituían una gran comunidad de usuarios y administradores. Es de anotar que esto incluía una buena contribución por parte de distribuidores de Linux.

Para coordinar el creciente número de parches, tres desarrolladores de software, Olivier Müller, Marc Delisle y Loïc Chapeaux, registraron el proyecto phpMyAdmin en SourceForge.net y continuó su crecimiento en 2001.

C#

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO. C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de una plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

El nombre C Sharp fue inspirado por la notación musical, donde '#' (sostenido, en inglés *Sharp*) indica que la nota (C es la nota do en inglés) es un semitono más alta, sugiriendo que C# es superior C/C++. Además, el signo '#' se compone de cuatro signos '+' pegados.

Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono-DotGNU, el cual genera programas para distintas plataformas como Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

Durante el desarrollo de la plataforma .NET, las bibliotecas de clases fueron escritas originalmente usando un sistema de código gestionado llamado Simple Managed C (SMC). En enero de 1999, Anders Hejlsberg formó un equipo con la misión desarrollar un nuevo lenguaje de programación llamado Cool (Lenguaje C orientado a objetos). Este nombre tuvo que ser cambiado debido a problemas de marca, pasando a llamarse C#. La biblioteca de clases de la plataforma .NET fue migrada entonces al nuevo lenguaje.

PHP

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

PHP fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término PHP.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o , Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante dominio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. El sitio web de Wikipedia está desarrollado en PHP. Es también el módulo Apache más popular entre los computadores que utilizan Apache como servidor web.

El gran parecido que posee PHP con lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Aunque todo en su diseño está orientado a facilitar la creación de sitios webs, es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de órdenes, de

la misma manera como Perl o Python pueden hacerlo; a esta versión de PHP se la llama PHP-CLI (Command Line Interface).

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite.

PHP también tiene la capacidad de ser ejecutado en la mayoría de sistemas operativos, tales como Unix y Microsoft Windows, y puede interactuar con los servidores web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponibles en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Ajax es una combinación de cuatro tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada(CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para interactuar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios. PHP es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico también utilizado en el método Ajax.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML pre formateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

A pesar de que el término “Ajax” fue creado en 2005 por Jesse James Garret, la historia de las tecnologías que permiten Ajax se remonta a una década antes con la iniciativa de Microsoft en el desarrollo de Scripting Remoto. Sin embargo, las técnicas para la carga asíncrona de contenidos en una página existente sin requerir recarga completa remontan al tiempo del elemento iframe (introducido en Internet Explorer 3 en 1996) y el tipo de elemento layer (introducido en Netscape 4 en 1997, abandonado durante las primeras etapas de desarrollo de Mozilla). Ambos tipos de elemento tenían el atributo src que podía tomar cualquier dirección URL externa, y cargando una página que contenga JavaScript que manipule la página paterna, pueden lograrse efectos parecidos al Ajax.

El Microsoft’s Remote Scripting (o MSRS, introducido en 1998) resultó un sustituto más elegante para estas técnicas de un applet Java el cual se puede comunicar con el cliente usando JavaScript. Esta técnica funcionó en ambos navegadores, Internet Explorer versión 4 y Netscape Navigator versión 4. Microsoft la utilizó en el Outlook Web Access provisto con la versión 2000 de Microsoft Exchange Server.

La comunidad de desarrolladores web, primero colaborado por medio del grupo de noticias Microsoft.public.scripting.remote y después usando blogs, desarrollaron una gama de técnicas de scripting remoto para conseguir los mismos resultados en diferentes navegadores. Los primeros ejemplos incluyen la biblioteca JSRS en el año 2000, la introducción a la técnica imagen/cookie en el mismo año y la técnica JavaScript bajo demanda (JavaScript on Demand) en 2002. En ese año, se realizó una

modificación por parte de la comunidad de usuarios al Microsoft's Remote Scripting para reemplazar el applet Java por XMLHttpRequest.

Frameworks de Scripting Remoto como el ARSCIF aparecieron en 2003 poco antes de que Microsoft introdujera Callbacks en ASP .NET.

Desde que XMLHttpRequest está implementado en la mayoría de los navegadores, raramente se usan técnicas alternativas. Sin embargo, todavía se utilizan donde se requiere una mayor compatibilidad, una reducida implementación, o acceso cruzado entre sitios web. Una alternativa, el Terminal SVG (basado en SVG), emplea una conexión persistente para el intercambio continuo entre el navegador y el servidor.

JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (cliente-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Una cuarta edición está en desarrollo e incluirá nuevas características tales como paquetes, espacio de nombres y definición explícita de clases.

JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente

quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador web Netscape Navigator en la versión 2.002 en diciembre de 1995. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación web.

“JavaScript” es una marca registrada de Oracle Corporation. Es usada con licencia por los productos creados por Netscape Communications y entidades actuales como la Fundación Mozilla.

Microsoft dio como nombre a su dialecto de JavaScript “JScript”, para evitar problemas relacionadas con la marca. JScript fue adoptado en la versión 3.0 de Internet Explorer, liberado en agosto de 1996, e incluyó compatibilidad con el Efecto 2000 con las funciones de fecha, una diferencia de los que se basaban en ese momento. Los dialectos pueden parecer tan similares que los términos “JavaScript” y “JScript” a menudo se utilizan indistintamente, pero la especificación es incompatible con la de ECMA en muchos aspectos.

Para evitar estas incompatibilidades, el World Wide Web Consortium diseñó el estándar Document Object Model (DOM, o Modelo de Objetos del Documento en español), que incorporan Konqueror, las versiones 6 de Internet Explorer y Netscape Navigator, Opera la versión 7, Mozilla Application Suite y Mozilla Firefox desde su primera versión.

En 1997 los autores propusieron JavaScript para que fuera adoptado como estándar de la European Computer Manufacturers Association ECMA, que a pesar de su nombre no es europeo sino internacional, con sede en Ginebra. En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de ECMAScript. Poco después como un estándar ISO.

Netscape introdujo una implementación de script del lado del servidor con Netscape Enterprise Server, lanzada en diciembre de 1994 (poco después del lanzamiento de JavaScript para navegadores web). A partir de mediados de la década de los 2000, ha habido una proliferación de implementaciones de JavaScript para el lado servidor. Node.js es uno de los notables ejemplos de JavaScript en el lado del servidor, siendo usado en proyectos importantes.

JavaScript se ha convertido en uno de los lenguajes de programación más populares en internet. Al principio, sin embargo, muchos desarrolladores renegaban el lenguaje porque el público al que va dirigido lo formaban publicadores de artículos y demás aficionados, entre otras razones. La llegada de Ajax devolvió JavaScript a la fama y atrajo la atención de muchos otros programadores. Como resultado de esto hubo una

proliferación de un conjunto de frameworks y librerías de ámbito general, mejorando las prácticas de programación con JavaScript, y aumentando el uso de JavaScript fuera de los navegadores web, como se ha visto con la proliferación de entornos JavaScript del lado del servidor. En enero de 2009, el proyecto CommonJS fue inaugurado con el objetivo de especificar una librería para uso de tareas comunes principalmente para el desarrollo fuera del navegador web.

HTML

HTML, siglas de HyperText Markup Language (lenguaje de marcado hipertextual), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para completar el texto con objetos tales como imágenes. El HTML se escribe en forma de “etiquetas”, rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir o hacer referencia a un tipo de programa llamado script, el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

HTML también sirve para referirse al contenido del tipo MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

La primera descripción de HTML disponible públicamente fue un documento llamado HTML Tags (Etiquetas HTML), publicado por primera vez en internet por Tim Berners-Lee en 1991. Describe 22 elementos que incluyen el diseño inicial y relativamente simple de HTML. Trece de estos elementos todavía existían en HTML 4.

Berners-Lee consideraba a HTML una ampliación de SGML, pero no fue formalmente reconocida como tal hasta la publicación de mediados de 1993, por la IETF, de una primera proposición para una especificación de HTML: el boceto Hypertext Markup Language de Berners-Lee y Dan Connolly, el cual incluía una Definición de Tipo de Documento SGML para definir la gramática. El boceto expiró luego de seis meses, pero fue notable por su reconocimiento de la etiqueta propia del navegador Mosaic usada para insertar imágenes sin cambio de líneas que reflejaba la filosofía del IETF de basar estándares en prototipos con éxito. De la misma manera, el boceto competidor de Dave Raggett HTML+ (Hypertext Markup Format) (Formato de Marcaje de Hipertexto), de finales de 1993, sugería estandarizar características ya implementadas, como las tablas.

HTML consta de varios componentes vitales, entre ellos los elementos y sus atributos, tipos de data y la declaración de tipo de documento.

Elementos: Los elementos son la estructuras son la estructura básica de HTML. Los elementos tienen dos propiedades básicas atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML. Un elemento generalmente tiene una etiqueta de inicio (por ejemplo, <nombre-de-elemento>) y una etiqueta de cierre (por ejemplo, </nombre-de-elemento>). Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas (por ejemplo, <nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>). Algunos elementos tales como
, no tienen contenido ni llevan una etiqueta de cierre.

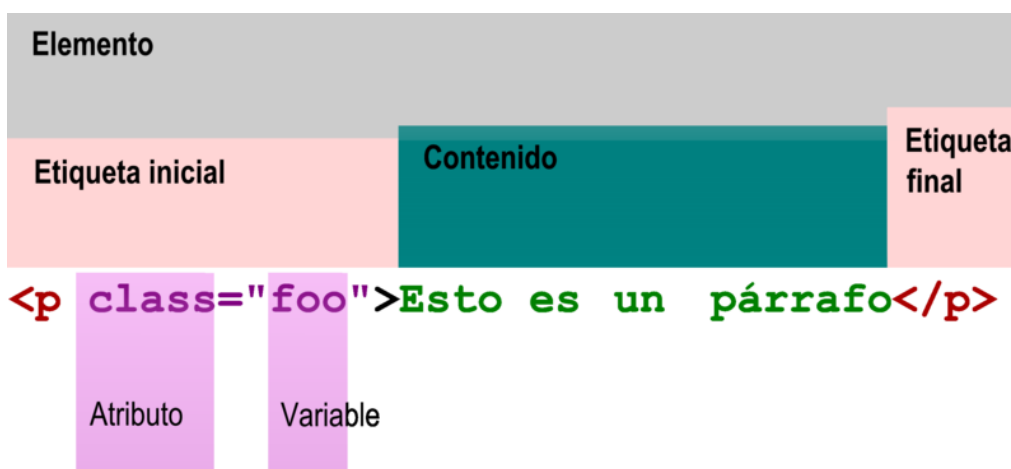


Figura 12: Estructura de una etiqueta HTML (extraída de <http://es.wikipedia.org/wiki/HTML>)

El marcado estructural describe el propósito del texto. Por ejemplo, <h2>Golf</h2> establece Golf como un encabezamiento de segundo nivel. El marcado estructural no define cómo se verá el elemento, pero la mayoría de los navegadores web han estandarizado el formato de los elementos. Puede aplicarse un formato específico al texto por medio de hojas de estilo en cascada.

El marcado presentacional describe la apariencia del texto, sin importar su función. Por ejemplo, negrita indica que los navegadores web visuales deben mostrar el texto en negrita, pero no indica qué deben hacer los navegadores web que muestran el contenido de otra manera (por ejemplo, los que leen el texto en voz alta). En el caso de negrita e <i>itálica</i>, existen elementos que se ven de la misma manera pero tienen una naturaleza más semántica: énfasis fuerte y se ven de la misma manera pero tienen una naturaleza más semántica: énfasis fuerte y énfasis. Es fácil ver cómo un lector de pantalla debería interpretar estos dos elementos. Sin embargo, son equivalentes a sus correspondientes elementos presentacionales: un lector de pantalla no debería decir

más fuerte el nombre de un libro, aunque éste esté en itálicas en una pantalla. La mayoría del marcado presentacional ha sido desechada con HTML 4.0, a favor de hojas de estilo en cascada.

El marcado hipertextual se utiliza para enlazar partes del documento con otros documentos o con otras partes del mismo documento. Para crear un enlace es necesario utilizar la etiqueta ancla `<a>` junto con el atributo `href`, que establecerá la dirección URL a la que apunta el enlace.

Atributos: La mayoría de los atributos de un elemento son pares nombre-valor, separados por un signo de igual "=" y escritos en la etiqueta de comienzo de un elemento, después del nombre de éste. El valor puede estar rodeado por comillas dobles o simples, aunque ciertos tipos de valores pueden estar sin comillas en HTML (pero no en XHTML). De todas maneras, dejar los valores sin comillas es considerado poco seguro. En contraste con los pares nombre-elemento, hay algunos atributos que afectan al elemento simplemente por su presencia.

CSS

Las hojas de estilo en cascada o (Cascading Style Sheets, o sus siglas CSS) hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "`<style>`".

CSS tiene una sintaxis muy sencilla, que usa unas cuantas palabras claves tomadas del inglés para especificar los nombres de sus selectores, propiedades y atributos. Una hoja de estilos CSS consta de 3 reglas. Cada regla consiste en uno o más selectores y un bloque de estilos con los estilos a aplicar para los elementos del documento que cumplan con el selector que les precede. Cada bloque de estilos se define entre dos llaves, y está formado por una o varias declaraciones de estilo con el formato propiedad:valor;.

En el CSS, los selectores marcarán qué elementos se verán afectados por cada bloque de estilo que les siga, pudiendo afectar a uno o varios elementos a la vez, en función de su tipo, nombre (name), ID, clase (class), posición dentro del Document Object Model, etcétera.

Para dar formato a un documento HTML, puede emplearse CSS de tres formas distintas:

1.- mediante CSS introducido por el autor de HTML

Un estilo de línea (online) es un método para insertar el lenguaje de estilo de página directamente dentro de una etiqueta HTML. Esta manera de proceder no es totalmente adecuada. El incrustar la descripción del formateo dentro del documento de la página Web, a nivel de código, se convierte en una manera larga, tediosa y poco elegante de resolver el problema de la programación de la página. Este modo de trabajo se podría usar de manera ocasional si se pretende aplicar un formateo con prisa, al vuelo. No es todo lo claro o estructurado que debería ser, pero funciona. Dado que los clientes de correo electrónico no soportan las hojas de estilos externas, y que no existen estándares que los fabricantes de clientes de correo respeten para utilizar CSS en este contexto, la solución más recomendable para maquetar correos electrónicos, es utilizar CSS dentro de los propios elementos (online).

Una hoja de estilo interna, que es una hoja de estilo que está incrustada dentro de un documento HTML, dentro del elemento <head>, marcada por la etiqueta <style>. De esta manera se obtiene el beneficio de separar la información del estilo del código HTML propiamente dicho. Se puede optar por copiar la hoja de estilo incrustada de una página a otra (esta posibilidad es difícil de ejecutar si se desea para guardar las copias sincronizadas). En general, la única vez que se usa una hoja de estilo interna, es cuando se quiere proporcionar alguna característica a una página Web en un simple fichero, por ejemplo, si se está enviando algo a la página Web.

Una hoja de estilo externa, es una hoja de estilo que está almacenada en un archivo diferente al archivo donde se almacena el código HTML de la página Web. Esta es la manera de programar más potente, porque separa completamente las reglas de formateo para la página HTML de la estructura básica de la página.

2.- Estilos CSS introducidos por el usuario que ve el documento, mediante un archivo CSS especificado mediante las configuraciones del navegador, y que sobrescribe los estilos definidos por el autor en una, o varias páginas web.

3.- Los estilos marcados “por defecto” por los user agent, para diferentes elementos de un documento HTML, como por ejemplo, los enlaces.

Arquitectura de la aplicación

La arquitectura de la aplicación, como se ha comentado con anterioridad, se apoya en el paradigma cliente-servidor, que se trata de un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da la respuesta. En este caso sería el servidor, donde se almacenaría toda la información y el cliente, solicitaría esa información y una vez recibida, se procesa para visualizarla.

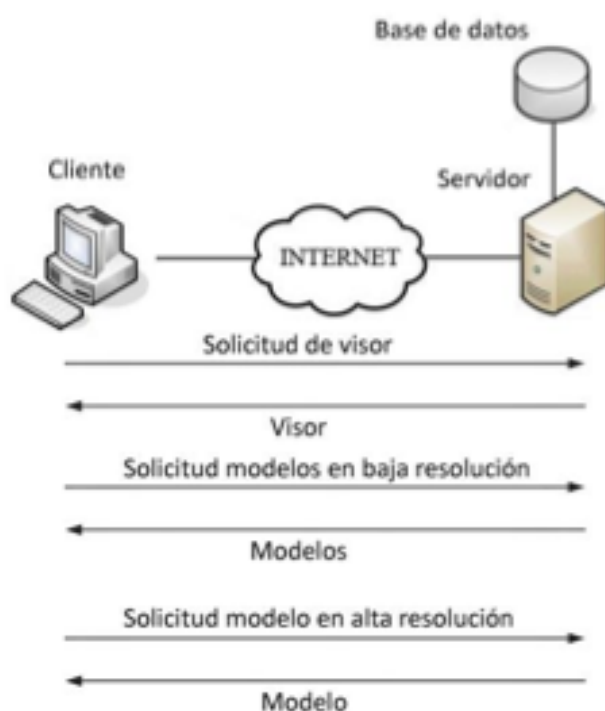


Figura 13: Diagrama Modelo Cliente-Servidor

La tecnología ha evolucionado mucho en los últimos años, por eso hoy en día prácticamente todo el mundo tiene acceso a internet y algún dispositivo móvil con conexión a la red, por ello, ésta arquitectura facilita mucho el acceso a la información. La elección de esta distribución conlleva que la base de datos requiera de varios ficheros PHP para que el visualizador, Front-End, pueda mostrar la información y que la herramienta de administración, Back-End pueda almacenar la información en la base de datos.

Habr  un fichero PHP por entidad de la base de datos que luego haya que mostrar en el visualizador,  stos ficheros, se conectar n con la base de datos mediante un usuario y contrase a, previamente definidos en la creaci n de  sta, en  stos ficheros habr  una consulta SQL por cada posible dato o informaci n que se pueda mostrar de la entidad en el visualizador.

Servidor

BBDD

Para el correcto funcionamiento de la aplicaci n es necesaria una base de datos para ello se crea una en phpmyadmin, tanto la estructura, como las tablas y los campos de  stas se han definido en colaboraci n con arquitectos y usuarios finales del proyecto. Una aplicaci n de estas caracter sticas, depende en gran medida de la informaci n que puede tratar, por eso  sta parte del proyecto es muy importante, la base de datos cuenta con un total de 30 tablas y de una gran cantidad de campos relacionados de la siguiente forma:

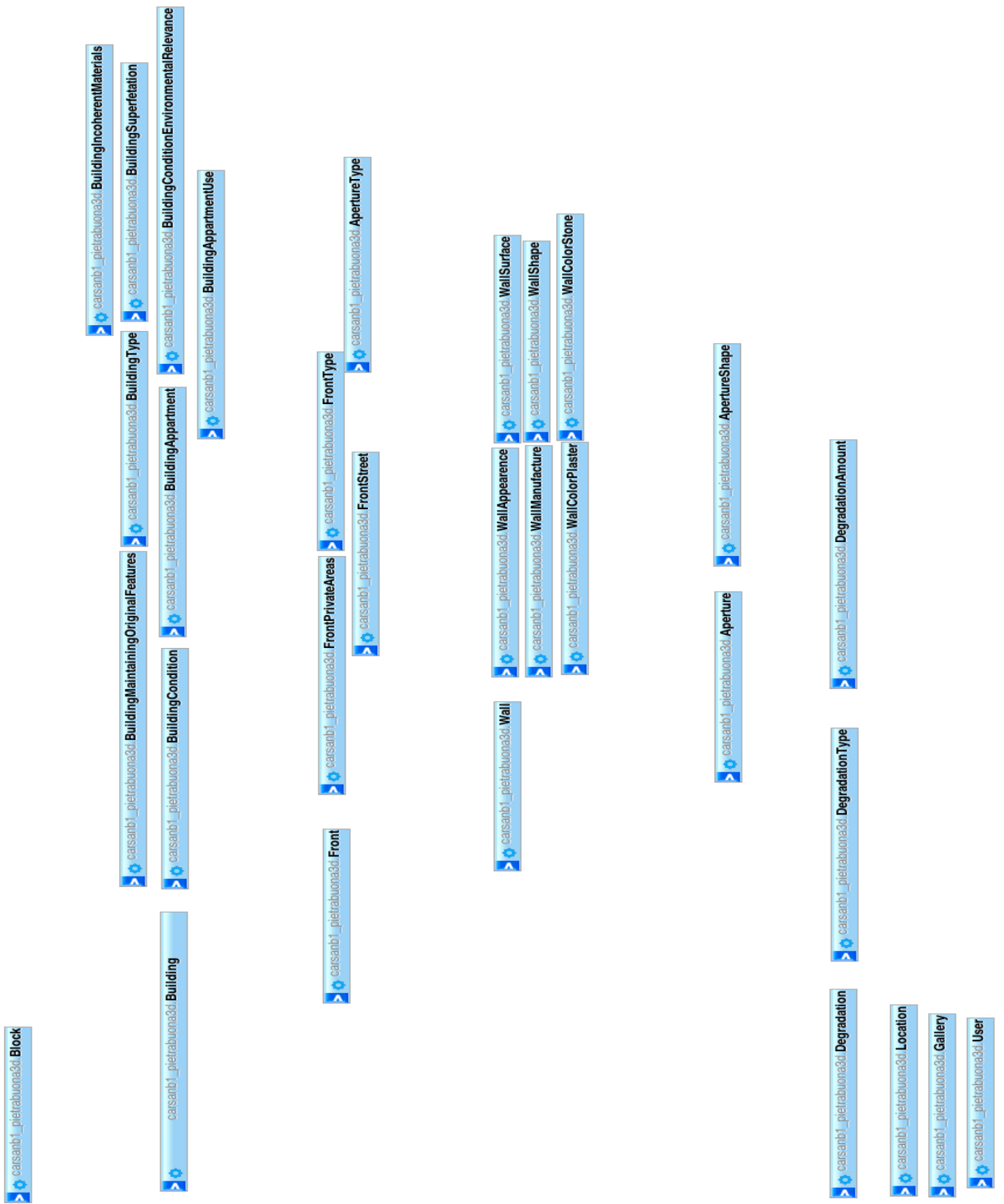


Figura 14: Diagrama de la base de datos

Como puede observarse, la estructura principal de los datos es una estructura hacia abajo, encontrando en la cima la tabla Block que correspondería a un bloque de edificios, el siguiente elemento sería Building que se relaciona con la tabla Block por el campo id_block ya que un Bloque de edificios puede tener uno o más edificios, a continuación tendríamos la tabla Front que se relaciona con una relación uno a muchos con Building por el campo id_building porque un edificio puede tener varios frentes. A su vez un frente puede tener uno o varios muros, de ahí que se relacione con la tabla Wall por el campo id_front y por último la tabla Aperture se uniría a Wall por id_wall por la misma razón que las anteriores.

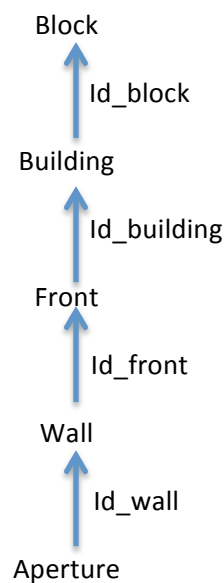


Figura 15: Diagrama de tablas principales

Por otro lado tenemos las tablas que complementan la información del bloque grueso de información son aquellas que contienen en su nombre alguna de las tablas del bloque central de información, estas tablas existen para los campos especiales en los que son tan diferentes de unas estructuras a otras que para permitir al usuario que defina sus propios campos era necesaria una estructuración de éste tipo. Para una mejor experiencia de usuario ha sido necesaria la tabla Location para que, como se explicará posteriormente, se pueda localizar dentro del modelo 3D la información referida a una parte de éste, también será necesaria la tabla Degradation y sus relacionadas para poder indicar el estado de conservación de los edificios, así como la tabla Gallery que es para guardar toda la información extra que el usuario requiera relacionada con una de las tablas principales, fotos, ficheros externos, etc. La gestión de usuarios de la herramienta de introducción de datos en la BBDD requiere una tabla más para almacenar el usuario y la contraseña de éstos.

Lógica interna

Back-End

Esta parte es la encargada de guardar la información en la base de datos, para ello se ha desarrollado esta aplicación en C# para evitar que el usuario se enfrente directamente con código de bases de datos, ya que el usuario final no requiere de conocimientos de bases de datos ni informáticos.

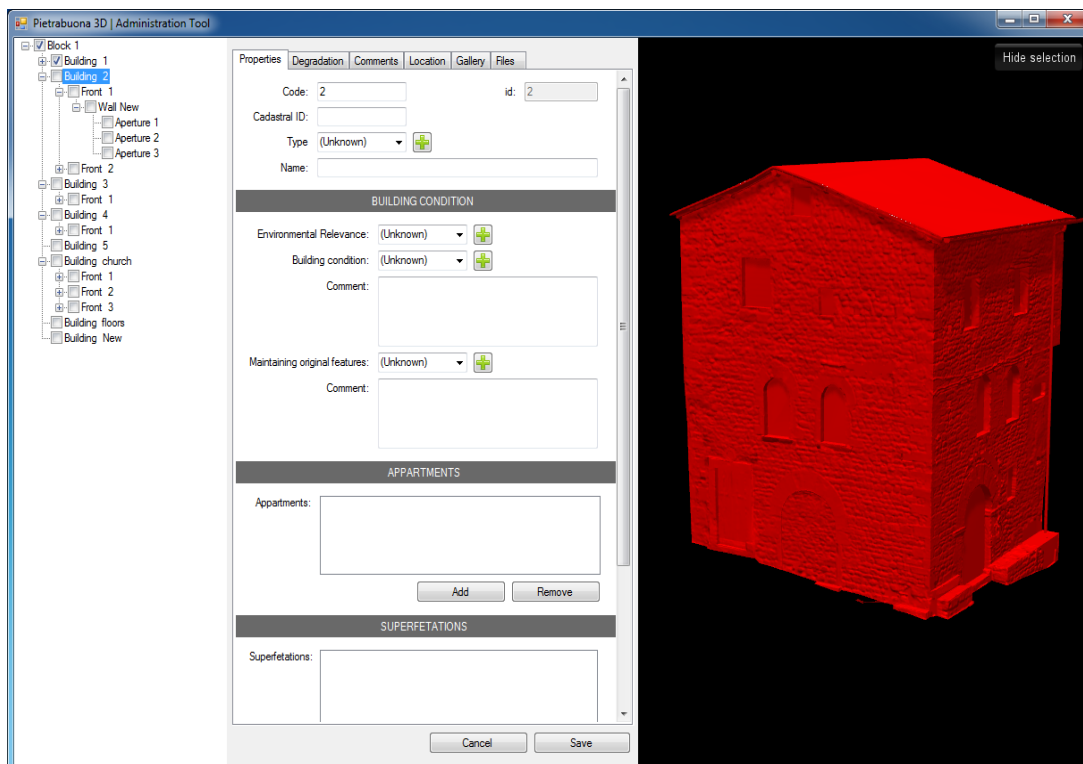


Figura 16: Pantalla del Administrador de datos

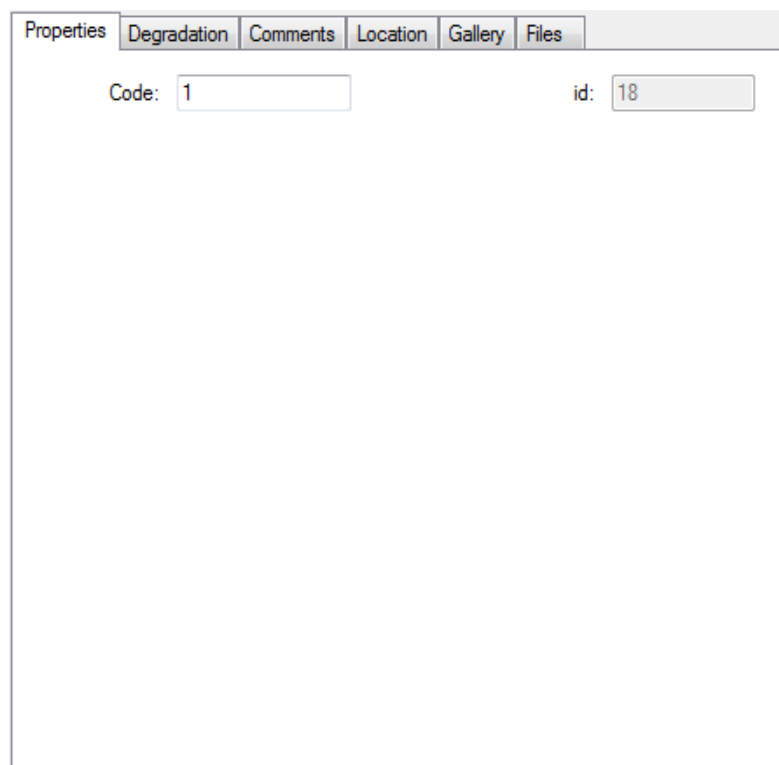
Como puede observarse, la aplicación está diferenciada en 3 partes, la primera es el árbol de navegación, la zona del centro es donde el usuario añade, modifica o elimina la información que se mostrará en el Front-End y la tercera parte, la de la izquierda, es donde se ve el modelo sobre el que se añade la información.

La parte de la derecha permite acceder a todos los elementos guardados en la base de datos y haciendo clic con el botón derecho se pueden añadir nuevos componentes.

Mediante éste árbol de navegación se puede acceder a la información relacionada con cada modelo almacenado en la base de datos, para la navegación por las distintas entidades principales de información se barajaron otras posibilidades como que

simplemente aparecieran los nombres, pero después de comentarlo con los usuarios finales se concluyó que ésta estructura era más cómoda ya que de un simple vistazo ya muestra la información de cómo están organizadas las entidades, los elementos almacenados aparecen en el árbol con un check box, esto es para mostrarlo luego en el front-end, si está activado el modelo se mostrará, esto mismo se puede conseguir con el botón que hay en la esquina superior derecha de la parte Unity donde se visualiza el modelo.

La segunda parte, la parte central, se divide en varias pestañas para no sobrecargar la pantalla y una mejor organización, la primera, es la de propiedades, esta pestaña es de las más importantes del Back-End ya que permite al usuario la introducción de la información del modelo seleccionado en ese momento, que cambia según el elemento que tengamos seleccionado en el árbol de navegación, así pues las propiedades del bloque serían el id, que no se podría modificar ya que es interno de la base de datos y el código de identificación.



The image shows a software interface with a tabbed menu at the top. The 'Properties' tab is active. Below the tabs, there are two input fields. The first is labeled 'Code:' and contains the number '1'. The second is labeled 'id:' and contains the number '18'. The rest of the page content is empty.

Figura 17: Propiedades del Bloque

La pestaña de propiedades de un edificio sería mucho más rica en información, por eso se dividiría en secciones, la primera sería la de información general, en la que tenemos el código de identificación, el id, el número catastral del edificio, el tipo, en el cual se dispone un botón con un símbolo “+” para poder añadir distintos tipos de edificio,

vivienda, iglesia, granja, etc., otro de los campos en esta categoría sería en nombre, por si el edificio tuviera uno.

El siguiente bloque de información que nos encontramos es el de condición del edificio, donde disponemos de la información relativa a la relevancia ambiental del edificio, la condición en la que se encuentra éste, estos dos parámetros se pueden agregar dándole al botón “+” y añadiendo la información pertinente. También se pueden añadir comentarios respecto a las condiciones del edificio, estructuras que se mantienen respecto al diseño original, por si por ejemplo se hubiera ampliado el edificio en otra época, los materiales y el estilo arquitectónico cambiaría, pues ahí quedaría reflejado, teniendo sus comentarios relativos a esas incongruencias arquitectónicas.

Otro bloque es el de los pisos, por si se quiere hacer algún comentario relativo a diferencias encontradas en los pisos, y por último materiales incoherentes en la estructura del edificio, como se puede observar, muchas de las posibilidades están orientadas a estudios arquitectónicos arqueológicos.

The image shows a software interface window titled 'Properties' with several tabs: 'Degradation', 'Comments', 'Location', 'Gallery', and 'Files'. The 'Properties' tab is active. It contains the following fields and sections:

- Code:** Input field with '2'.
- id:** Input field with '2'.
- Cadastral ID:** Input field.
- Type:** Dropdown menu with '(Unknown)' selected and a '+' icon.
- Name:** Input field.
- BUILDING CONDITION:** A section header followed by:
 - Environmental Relevance:** Dropdown menu with '(Unknown)' and a '+' icon.
 - Building condition:** Dropdown menu with '(Unknown)' and a '+' icon.
 - Comment:** Text area.
 - Maintaining original features:** Dropdown menu with '(Unknown)' and a '+' icon.
 - Comment:** Text area.
- APPARTMENTS:** A section header followed by:
 - Appartments:** Text area.
 - Add** and **Remove** buttons.
- SUPERFETATIONS:** A section header followed by:
 - Superfetations:** Text area.
 - Add** and **Remove** buttons.
- INCOHERENT MATERIALS:** A section header followed by:
 - Incoherent Materials:** Text area.
 - Add** and **Remove** buttons.

At the bottom of the window are **Cancel** and **Save** buttons.

Figura 18: Propiedades del Edificio

La pestaña de propiedades del frente hace referencia a la cara del edificio, ya que un edificio puede tener más de una cara, esta pestaña, como la anteriormente explicada,

está dividida en bloques de datos para facilitar su localización, primero los datos generales, el código y el id, también si es frente principal del edificio y tipo, en este último, de le da la opción al usuario para añadir sus propios tipos.

Una vez visto esto, pasamos a la parte de las plantas, en este bloque de trata el número de pisos máximo y mínimo que debía tener el edificio y si el suelo de la vivienda estaba parcialmente a un nivel más bajo que la calle. El siguiente bloque indica hacia dónde está orientada esa cara del edificio, puede ser una calle, con la posibilidad de añadir el nombre de la calle, una zona abierta, o un área privada, dando la posibilidad de añadirla, esta parte está para las caras del edificio que daban a jardines interiores o zonas similares.

The image shows a software window titled 'Properties' with several tabs: 'Properties', 'Degradation', 'Comments', 'Location', 'Gallery', and 'Files'. The 'Properties' tab is selected. The window contains the following fields and sections:

- Code:** 1
- id:** 30
- Main Front:**
- Type:** (Unknown) with a plus icon for adding new types.
- NUMBER OF FLOORS** section:
 - Number of floors (max):** 1
 - Number of floors (min):** 1
 - First floor partially underground:**
- FACING TO** section:
 - Facing to:** Radio buttons for 'Street (-1) (Unknown)', 'Land' (selected), and 'Private areas: (Unknown)' with a plus icon.
- ACCESS** section:
 - Number:** Radio buttons for 'With 1' and 'Without' (selected).

At the bottom of the window are 'Cancel' and 'Save' buttons.

Figura 19: Propiedades del Frente


Éstas serían las propiedades de la pared, en este caso, hay 2 distintas posibilidades dependiendo del material, en ambos casos está el código y la id, y luego hay un combo box donde se puede elegir el material, piedra o yeso, si hemos elegido yeso, podemos tratar el estilo de fabricación y el color, en ambas propiedades podemos añadir la que queramos y luego elegirla. En el caso de elegir piedra como material las propiedades cambian y disponemos de apariencia, elaboración de los ladrillos, color y otras propiedades.

Properties | Degradation | Comments | Location | Gallery | Files

Code: id:

Material:

PLASTER

Manufacture: 


Color: 


Figura 20: Propiedades de la pared de Yeso

Properties | Degradation | Comments | Location | Gallery | Files


Code: id:


Material:


STONE

Appearance: 

Brick elaboration

Surface: 

Shape: 

Number of Listatura: 


Color: 

Figura 21: Propiedades de la pared de Piedra

Si seleccionamos una apertura, tenemos las siguientes propiedades, las comunes, el código y la id y en las específicas el tipo, la planta donde se encuentra y la forma, en este caso se entiende como apertura cualquier hueco creado voluntariamente para el edificio, encontrándose en este tipo las puertas o las ventanas

The image shows a software interface window titled 'Properties'. It has several tabs: 'Degradation', 'Comments', 'Location', 'Gallery', and 'Files'. The 'Properties' tab is selected. Inside the window, there are several input fields and dropdown menus. 'Code' is a text box containing '1'. 'id' is a text box containing '47'. 'Type' is a dropdown menu showing '(Unknown)' with a green '+' icon to its right. 'Shape' is a dropdown menu showing '(Unknown)' with a green '+' icon to its right. 'Floor' is a spinner box containing '0'. At the bottom of the window, there are two buttons: 'Cancel' and 'Save'.

Figura 22: Propiedades de la Apertura

Cada uno de estos elementos se ha tratado internamente como una clase distinta, y las propiedades son los atributos de ésta, hay propiedades especiales en las que se deja la posibilidad de que el usuario agregue tipos, es el caso de las propiedades que tienen a su lado un símbolo “+” de color verde, en este caso los atributos de esta clase son listas, ya que se puede dar el caso de que haya más de uno.

La siguiente pestaña de esta zona del programa es la degradación, en este caso son todas iguales, ya que la información que se guarda aquí es el tipo de degradación, la cantidad y luego hay un apartado para guardar comentarios relativos a ésta, cada parte del modelo puede encontrarse en distintos estados de conservación, es decir, una iglesia puede tener un muro en buen estado, pero el campanario estar en otro

totalmente distinto, por eso se da la opción de que añadir información del deterioro a cada elemento.

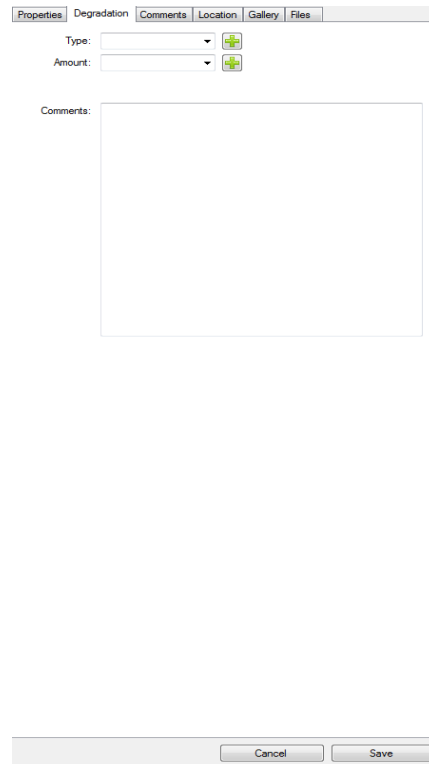


Figura 23: Pestaña Degradation

También se puede encontrar una pestaña de comentarios, para añadir texto libre a cada parte del edificio, y una galería de imágenes o documentos, para dar la opción de guardar fotografías de cada parte del edificio para resaltar frescos que no se pueden escanear o información relevante, así como otros tipos de ficheros como artículos PDF o similares.

Otra de las pestañas importantes es la pestaña localización, esta pestaña es para controlar los parámetros de visualización del modelo, en la parte del Front-End, lo primero es la posición donde se colocará la cámara al seleccionar el elemento en la parte de visualización, la segunda parte es donde se encuentra dentro del mismo modelo el elemento seleccionado, esto se puede seleccionar añadiendo las coordenadas como punto de interés o seleccionando una parte del modelo, por último se puede seleccionar la máscara que es la capa roja que nos marca la parte del modelo sobre la que estamos observando la información. Toda esta información servirá posteriormente para el visualizador, uno de los problemas de mostrar la información en 3 dimensiones que localizar la información dentro del modelo, si estás viendo una cara de un edificio, como éstos tienen más de una, hay que buscar una forma de mostrarle al usuario a que parte del edificio está haciendo referencia la información

mostrada, para ello se ha pensado que cuando se seleccione un elemento en el árbol de navegación, la cámara se posicione sola mostrando el elemento que ha sido elegido, y para ello es el usuario, al introducir la información el que ha de posicionar la cámara. Por otro lado, cuando el elemento es una parte más pequeña, habría que resaltarla aún más de ahí que se use una máscara de color rojo para resaltar por encima de los otros elementos la parte específica sobre la que se está tratando.

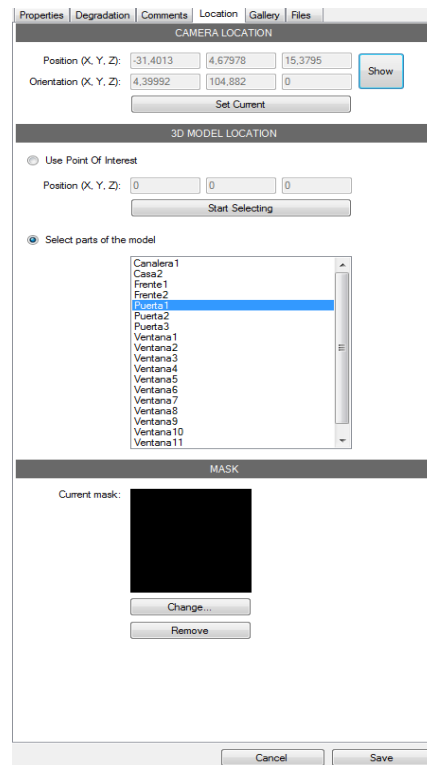


Figura 24: Pestaña Location

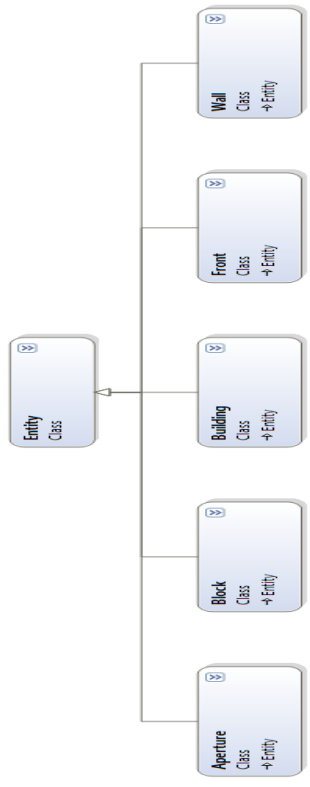
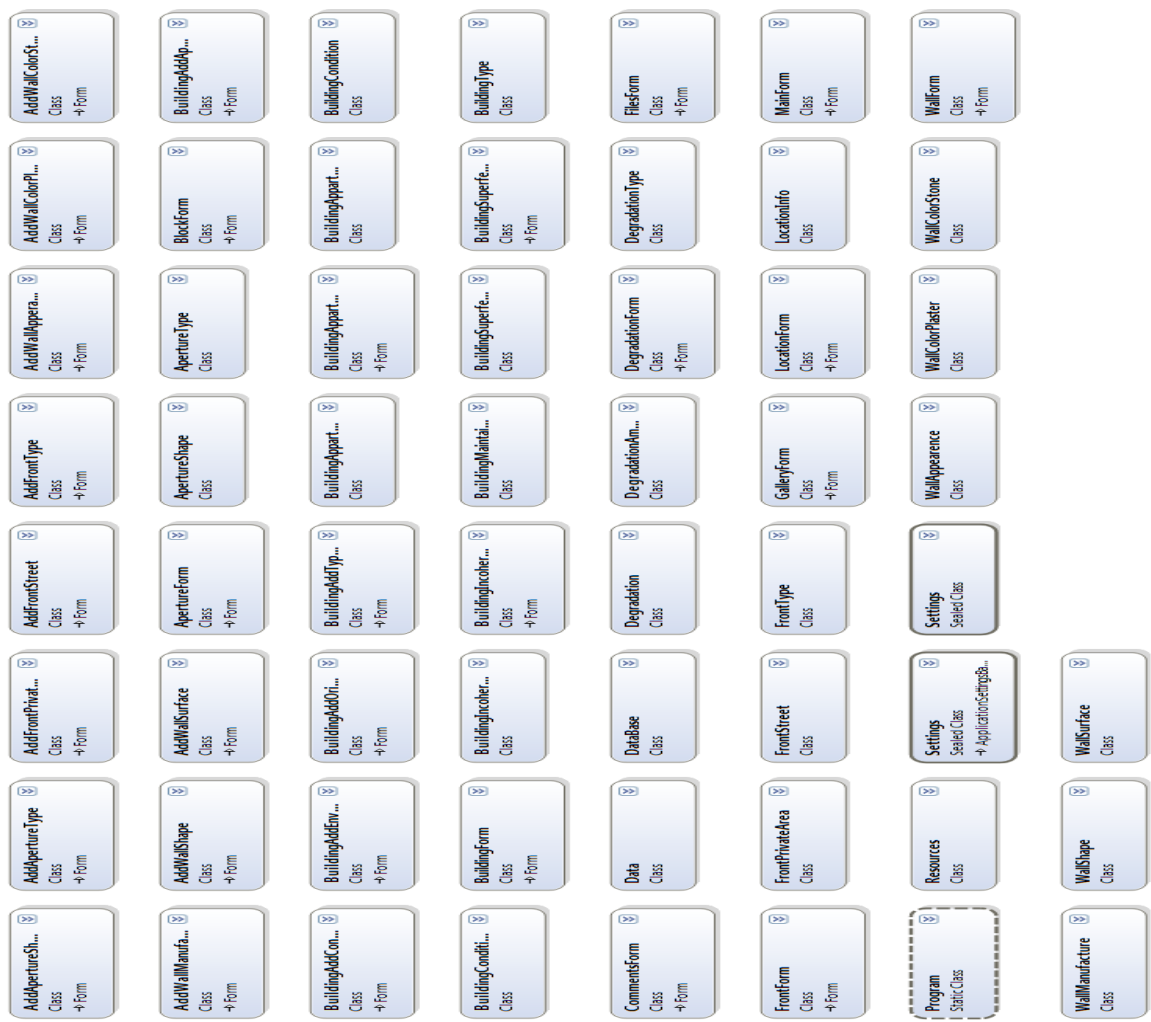


Figura 25: Diagrama de Clases

Front-End

Esta parte es la encargada de mostrar la información al usuario, para permitir una facilidad mayor a la hora de acceder a los datos, se ha creado una página web desde la que acceder a la información almacenada en la base de datos por medio del Back-End.

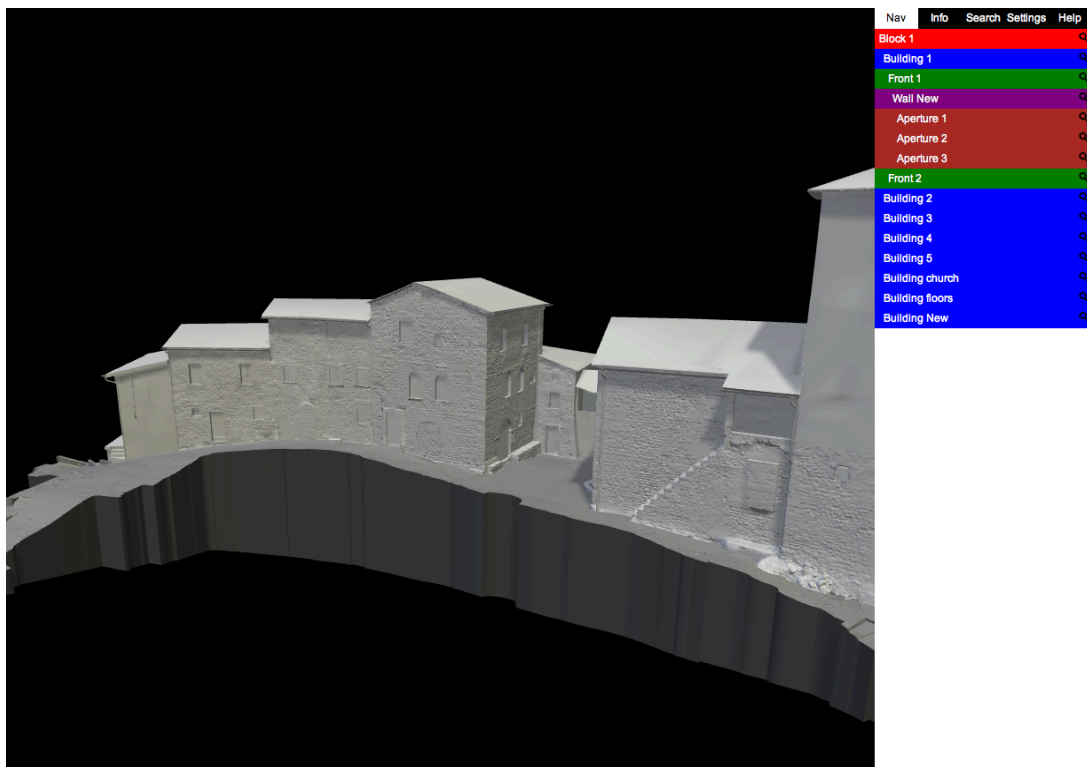


Figura 26: Pantalla del Front-End

La pantalla se divide en dos partes bien diferenciadas, la parte de la izquierda, es la parte de visualización de los modelos almacenados en la BBDD. Esta parte está creada con Unity3D e insertada en la web, el modelo se puede rotar y mover con el ratón para una correcta visualización del modelo, los modelos que se muestren serán los que previamente se han activado desde la herramienta de introducción de la información.

Esta parte realizada en Unity3D de momento solo permite el movimiento del modelo, se ha desarrollado en C# asignando scripts a los elementos de la escena, la cámara y el modelo, usando esta misma tecnología se podrían desarrollar varias herramientas para medir distancias, superficies, volúmenes, etc.

```

function GetUnity() {
    if (typeof unityObject != "undefined") {
        return unityObject.getObjectById("unityPlayer");
    }
    return null;
}
if (typeof unityObject != "undefined") {
    var params = {disableContextMenu: true};
    unityObject.embedUnity("unityPlayer", "Pietrabuona.unity3d", "100%", "100%", params);
}

```

Código 1: Inserción del visualizador Unity3D

Para la navegación por el modelo se usa el ratón, la cámara inicialmente se sitúa a una distancia para que se pueda tener todo el modelo en el espacio de la visión, con el botón izquierdo del ratón podemos rotar la cámara sobre el centro del modelo, manteniendo el botón derecho pulsado podemos mover la cámara en los ejes X e Y.

La parte derecha de la pantalla es para navegar por los elementos que forman el modelo, para ello se dispone de unas pestañas en las que se puede acceder a los servicios que ofrece la web y navegar por la información.

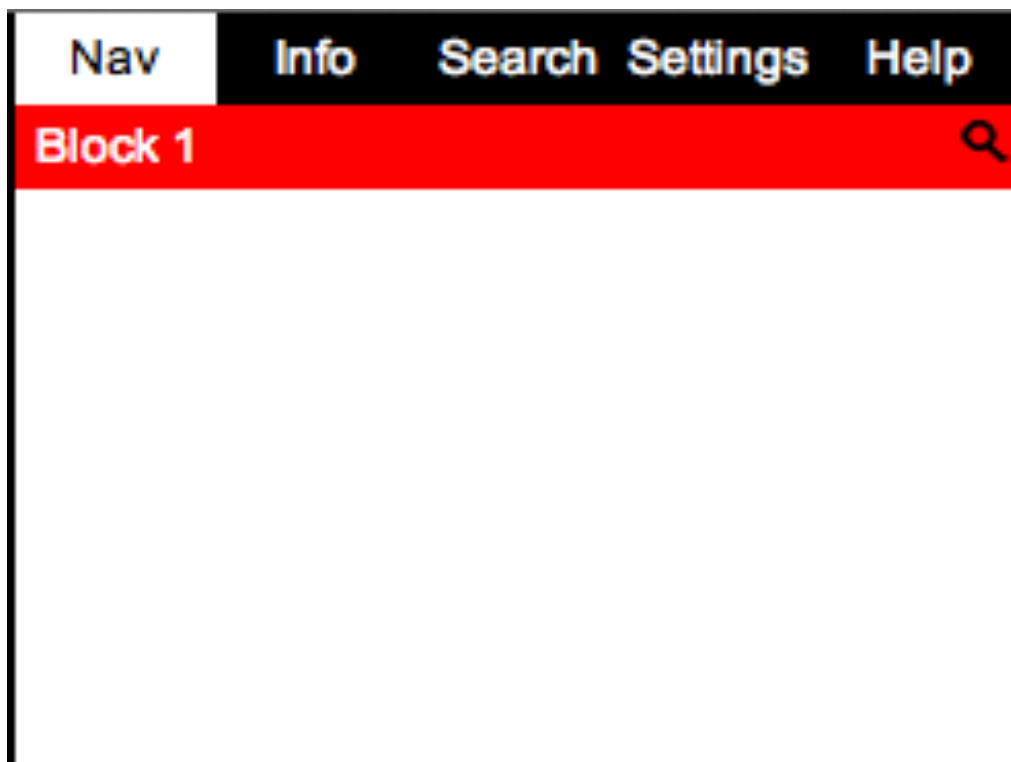


Figura 27: Navegación en el Front-End

Las pestañas nos ofrecen la posibilidad de navegar por los elementos del modelo en la pestaña “Nav”, nos muestran la información correspondiente al elemento seleccionado en “info”, podemos buscar elementos por código o nombre en la pestaña “Search”, cambiar algunos parámetros del visualizador en la pestaña “Settings” o consultar la ayuda en “Help”.

```
echo '<div id="tab" class="tab0" onClick="ShowNavigationTree()">Nav</div>';
echo '<div id="tab" class="tab1" onClick="ShowEntityInfo(-1, -1)">Info</div>';
echo '<div id="tab" class="tab2" onClick="ShowSearch()">Search</div>';
echo '<div id="tab" class="tab3" onClick="ShowSettings()">Settings</div>';
echo '<div id="tab" class="tab4" onClick="ShowHelp()">Help</div>';
```

Código 2: Tablas en la Web

El primer elemento que veremos en la pestaña “Nav” siempre será un Block, ya que el bloque será el elemento básico para nuestros modelos, un Block correspondería a una manzana de viviendas en nuestro caso.

```
$queryBlock = mysql_query("SELECT * FROM `Block` ORDER BY `order`");
while($block = mysql_fetch_array($queryBlock)) {
    echo '<div id="blockButton" <span id="entitySpan" onClick="NodeDisplay(\'block\' . $block["id"] . \'')">Block ' . $block["code"] . ' </span><div id="zoom"
onClick="ShowEntityInfo(\'Block\', ' . $block["id"] . ')"></div></div>';
    echo '<div id="block" . $block["id"] . ' . ' style="display:none">';
    $queryBuilding = mysql_query("SELECT * FROM `Building` WHERE `id_block` = " . $block["id"] . " ORDER BY `order`");
    while($building = mysql_fetch_array($queryBuilding)) {
        echo '<div id="buildingButton" <span id="entitySpan" onClick="NodeDisplay(\'building\' . $building["id"] . \'')">Building ' . $building["code"] . ' </span><div id="zoom"
onClick="ShowEntityInfo(\'Building\', ' . $building["id"] . ')"></div></div>';
        echo '<div id="building" . $building["id"] . ' . ' style="display:none">';
        $queryFront = mysql_query("SELECT * FROM `Front` WHERE `id_building` = " . $building["id"] . " ORDER BY `order`");
        while($front = mysql_fetch_array($queryFront)) {
            echo '<div id="frontButton" <span id="entitySpan" onClick="NodeDisplay(\'front\' . $front["id"] . \'')">Front ' . $front["code"] . ' </span><div id="zoom"
onClick="ShowEntityInfo(\'Front\', ' . $front["id"] . ')"></div></div>';
            echo '<div id="front" . $front["id"] . ' . ' style="display:none">';
            $queryWall = mysql_query("SELECT * FROM `Wall` WHERE `id_front` = " . $front["id"] . " ORDER BY `order`");
            while($wall = mysql_fetch_array($queryWall)) {
                echo '<div id="wallButton" <span id="entitySpan" onClick="NodeDisplay(\'wall\' . $wall["id"] . \'')">Wall ' . $wall["code"] . ' </span><div id="zoom"
onClick="ShowEntityInfo(\'Wall\', ' . $wall["id"] . ')"></div></div>';
                echo '<div id="wall" . $wall["id"] . ' . ' style="display:none">';
                $queryAperture = mysql_query("SELECT * FROM `Aperture` WHERE `id_wall` = " . $wall["id"] . " ORDER BY `order`");
                while($aperture = mysql_fetch_array($queryAperture)) {
                    echo '<div id="apertureButton" <span id="entitySpan" onClick="NodeDisplay(\'aperture\' . $aperture["id"] . \'')">Aperture ' . $aperture["code"] . ' </span><div
id="zoom" onClick="ShowEntityInfo(\'Aperture\', ' . $aperture["id"] . ')"></div></div>';
                    echo '<div id="aperture" . $aperture["id"] . ' . ' style="display:none">';
                }
            }
        }
    }
}
```

Código 3: Pestaña de navegación

El siguiente elemento en el que se divide el Block es el Building, ya que una manzana se divide en edificios, éstos se dividen en Fronts que serían caras de los edificios que a su vez se dividen en Walls, muros, ya que una cara de un edificio puede estar formada por varios muros, los muros pueden tener Apertures que pueden ser ventanas, puertas, etc.



Figura 28: Sistema de navegación

El efecto de despliegue hacia abajo se consigue usando la librería jQuery de JavaScript, asignando a cada elemento desplegable una id y usando el método slideUp.

```
function NodeDisplay(id) {
    if($("#" + id).css('display')== 'none') $("#" + id).slideDown("fast");
    else $("#" + id).slideUp("fast");
}
```

Código 4: Efecto de despliegue

Como se puede observar, detrás de cada elemento está el icono de una lupa, clicando en él, se accede directamente a la información relativa a ese elemento, cada parte del edificio tiene unas características y unas propiedades diferentes.

Nav	Info	Search	Settings	Help
Front New				
ID 41				
The material of this wall is Stone				
The listatura is 0				
Wall appearance: Unknown				
Wall surface: Unknown				
Wall shape: Unknown				
The color of the stone is: Unknown				
The manufacture of the stone is: Unknown				

Figura 29: Pestaña de Información

```

$res = mysql_query("SELECT * FROM `Front` WHERE `id`=" . $id . " LIMIT 1");
while($row = mysql_fetch_array($res)) {
    echo '<div id="InfoTitle">Front ' . $row["code"] . '</div>';
    echo '<div id="InfoContents">';
    echo '<p>ID ' . $row["id"] . '</p>';
    if($row["mainFront"]==1) echo '<p>This is the main front</p>';
    echo '<p> The maximum number of floors is ' . $row["max_floors"] . '</p>';
    echo '<p> The minimum number of floors is ' . $row["min_floors"] . '</p>';
    if($row["partially_underground"]==1) echo '<p>This is front is partially underground</p>';
    echo '<p> Numbre of the front ' . $row["number"] . '</p>';

    // this part is for the subtables
    $stype = "Unknown";
    $stypeQuery = mysql_query("SELECT * FROM `FrontType` WHERE id = " . $row["id_front_type"]);
    if(mysql_num_rows($stypeQuery) > 0) {
        $rowfrontType = mysql_fetch_array($stypeQuery);
        $stype = $rowfrontType["text"];
    }
    echo '<p>Front type: ' . $stype . '</p>';

    $street = "Unknown";
    $streetQuery = mysql_query("SELECT * FROM `FrontStreet` WHERE id = " . $row["id_front_street"]);
    if(mysql_num_rows($streetQuery) > 0) {
        $rowfrontStreet = mysql_fetch_array($streetQuery);
        $street = $rowfrontStreet["text"];
    }
    echo '<p>Front street: ' . $street . '</p>';

    $private_area = "Unknown";
    $private_areaQuery = mysql_query("SELECT * FROM `FrontPrivateAreas` WHERE id = " . $row["id_front_private_area"]);
    if(mysql_num_rows($private_areaQuery) > 0) {
        $rowfrontPrivateArea = mysql_fetch_array($private_areaQuery);
        $private_area = $rowfrontPrivateArea["text"];
    }
    echo '<p>Front private area: ' . $private_area . '</p>';

    if(strlen($row["comments"]) > 0) {
        echo '<p>Comments: ' . $row["comments"] . '</p>';
    }
    echo '</div>';
}

```

Código 5: Pestaña de información

Para crear el sistema de pestañas se ha usado HTML, CSS, PHP y la librería de JavaScript jQuery, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web, con PHP se hacen las consultas a la BBDD y se muestran con HTML, luego con CSS se le da estilo y con jQuery se le da la funcionalidad, es el funcionamiento estándar de las páginas web, con JavaScript se introduce también el módulo Unity3D con el que se visualizan los modelos.

Casos de uso

Al finalizar tenemos tres elementos clave, desarrollados para facilitar el trabajo de los arquitectos especializados en arquitectura arqueológica, por un lado la base de datos, totalmente indispensable para almacenar toda la información necesaria así como los modelos, después de escanearlos en 3d, la base de datos ha sido desarrollada desde un principio con ayuda de arquitectos y usuarios finales del producto.

Por otro lado tenemos las herramientas con las que los usuarios interactuarán, el Back-End, donde un usuario con rol de moderador podrá introducir los datos en la base de datos, esta herramienta es una aplicación desarrollada en Visual Studio en el lenguaje de programación C#, para evitar que cualquiera pueda modificar la información o estropear los modelos se requerirá de usuario y contraseña suministrados al crear la base de datos, con estos datos ya se puede empezar a analizar el entorno y subir información rellenando las casillas que hay disponibles.

Luego la parte del Front-End estará accesible desde cualquier parte ya que está desarrollada con tecnología web, para poder compartir la información de la base de datos de una manera cómoda y simple solamente compartiendo la URL de ésta, dando acceso a los datos y modelos ya sea con fines didácticos o informativos. Con la tecnología de hoy en día se podría acceder desde cualquier parte siempre y cuando se tenga conexión a internet, facilitando el intercambio de información ya que haría falta menos gente en el lugar del estudio arquitectónico al tener sólo que rellenar los datos del Back-End y otro conjunto de expertos podrían estar en otro lugar analizando los datos recolectados todo ello conllevaría un menor coste en las expediciones y la posibilidad de compartir la información con más gente.

El producto final está pensado para el análisis de estructuras arquitectónicas, permitiendo ahorrar costes a la hora de estudiar los restos arquitectónicos dando lugar a que los equipos que viajen al lugar donde se encuentran sea menos numeroso, pero aumentando la posibilidad de compartir y por tanto aumentar los equipos de investigación de esas estructuras.

Otro posible uso puede ser para las inmobiliarias a la hora de presentar los pisos en venta, al permitir introducir los modelos 3D y la información relativa, se podrían crear modelos de los pisos que oferta la inmobiliaria y añadir la información relativa a la habitación en la que se encuentra en ese momento ya que el usuario "cliente" podría hacer un tour por el piso en 3D.

Exportando el posible uso anterior a una zona más amplia, los ayuntamientos también podrían crear modelados 3D de sus estructuras arquitectónicas más emblemáticas, para facilitar al usuario, localizaciones y datos sobre éstas a la hora de hacer una visita turística.

Si nos centramos en el uso arquitectónico, se podrían crear modelos de edificios que se vayan a proyectar para comprobar el impacto ambiental o estético que podrían generar en la localización final de éste, permitiendo ahorrar costes ya que se evitaría la maquetación, con posibles ampliaciones se podría además simular como afectarían los años para comprobar posibles materiales de construcción a la hora de evitar humedades, erosiones, etc.

Dadas sus características y posibilidades un GIS como el presentado puede ayudar en la planificación urbanística de las ciudades, a la hora de levantar un nuevo edificio se podría mirar el impacto que tendría en la ciudad tanto estéticamente como funcionalmente.

A la hora de construir un edificio también se podría usar este GIS para comprobar los mejores materiales de construcción, aislamiento etc. Teniendo en cuenta la localización del edificio se podrían hacer simulaciones de diversas calidades de materiales para aislar del frío y del calor, ya que dependiendo de la posición y los edificios colindantes se puede estudiar las horas de luz, paredes y ventanas a las que más les de la luz y con ello elegir el material con mejor calidad/precio a la hora de construir el edificio. Lo mismo ocurre con el sonido, ya que dependiendo de la localización y de las calles que tenga cerca el edificio habrá que aislar más o menos del ruido, no es lo mismo tener un paso peatonal que unas vías de tranvía al lado del edificio a la hora de elegir materiales de construcción.

Posibles ampliaciones

El producto final como se puede apreciar es una versión básica de las tres partes que forman el proyecto, en la base de datos se podrían añadir múltiples tablas y campos para hacer más específico o más amplio el programa para responder a otras necesidades del usuario final, en el caso que nos ocupa ha sido pensado para satisfacer las necesidades de un equipo de arquitectos que está estudiando un pequeño pueblo en Italia llamado Pietrabuona, por ello las tablas y campos están pensados para éste equipo, aún así, para todo el que quiera hacer un estudio del mismo estilo de cualquier arquitectura, serviría la misma estructura de la base de datos.

El Back-End, al ser la herramienta para introducir la información está estrechamente ligado al diseño de la base de datos, por que lo cualquier cambio en ésta conllevaría una modificación en el programa de introducción de datos, pero se podrían añadir los campos para añadir toda la información que se deseara.

Por otro lado tenemos el Front-End que es la parte más sencilla en este momento y que más juego puede dar de cara a posibles expansiones. En la versión actual del visualizador, lo único que hace es mostrar la información almacenada en la base de datos relacionada con los modelos y permitir una visualización de éstos. Se puede apreciar que las posibilidades son muchas a partir de aquí, ya que se podría aprovechar toda la información con la que trabaja se podría procesar y desarrollar para llegar a conclusiones, de muy diversos tipos.

Se podría perfeccionar la parte del visualizador de modelos, creando paseos virtuales por los modelos introduciendo coordenadas por las que podemos querer llevar al usuario y luego mover la cámara por las coordenadas seleccionadas y así añadiendo las coordenadas y un botón en la parte del visualizador, se podría montar un paseo virtual por donde el administrador desee. También se puede dar más movilidad a la cámara permitiendo que tenga más movimientos asociados a teclas, imitando a los videojuegos usando las teclas w, s, d y a para desplazarnos arriba, abajo, derecha y abajo o usando las flechas del teclado.

Otra ampliación podría ser usar las coordenadas de las estructuras para colocar la cámara en la posición almacenada al seleccionarla en la pestaña de navegación, permitiendo al usuario conocer con exactitud el elemento al que se refiere la información de la pestaña "Info".

En la pestaña se "Search" se podría poner un buscador de características, para permitir al usuario buscar por ejemplo, estructuras de un año en concreto, de un determinado material, etc.

La pestaña se “Settings” está aún en construcción, por lo que no tiene aún funcionalidad definida, por ello se podría añadir la posibilidad de cambiar colores u opciones de visualización.

En la pestaña de “Help” se podrían poner las explicaciones pertinentes para que el usuario supiera manejar el visualizador como el funcionamiento de la cámara, las teclas para el movimiento, como usar el buscador, etc.

Con los datos ya almacenados, podemos plantearnos otras ampliaciones enfocados a ampliar el poder de análisis de la información guardada en la base de datos. Se pueden dividir las posibles ampliaciones respecto a que uso se quiere dar a esta información, al estar pensado para arquitectura arqueológica vamos a centrarnos en dos disciplinas, mantenimiento y restauración

Mantenimiento de estructuras:

Ya que entre los datos almacenados se encuentran los materiales usados para la construcción de las estructuras, con poca información más se pueden hacer estudios de la erosión de las ruinas.

Dentro del ambiente exógeno, uno de los procesos más importantes que tienen lugar, debido a la dinámica superficial del planeta, es el desgaste físico y químico que sufren las rocas bajo la acción de los agentes atmosféricos. Asociado a este proceso de transporte de los productos de la erosión (fragmentos de rocas, minerales, sales) por los mismos agentes que producen los fenómenos de erosión: el agua, y el viento.

Los procesos erosivos tienen lugar como consecuencia de tres grupos de fenómenos:

- 1.- Los de carácter físico, ligados a cambios de temperatura, o de estado físico del agua (cristalización de hielo en grietas).
- 2.- Los de tipo químico (disolución de minerales, hidrólisis de éstos, cristalización de sales).
- 3.- Los de tipo biológico (acción de determinados microorganismos, como las bacterias, líquenes, o de raíces de plantas).

Como resultado, las estructuras de la superficie terrestre al quedar sometidas a presiones y temperaturas reaccionan con el entorno, lo que introduce un desequilibrio. Esto da lugar a su fragmentación y a la salida de determinados componentes químicos.

Como ya se ha indicado, la erosión tiene lugar mediante tres grupos de mecanismos: físicos, químicos y biológicos, que en general se combinan, con mayor o menor importancia de unos u otros en función de un factor primordial: el clima, que condiciona a su vez la disponibilidad de agua, de vegetación, las temperaturas medias, sus oscilaciones. Estos factores influyen en la degradación a la intemperie de cualquier sólido. Por ello, hay climas que favorecen la preservación de las rocas, y climas bajo los que se produce una muy intensa meteorización, así como la rápida descomposición de cualquier resto orgánico.

Los factores que influyen en la erosión son el clima, que tiene una influencia fundamental, ya que controla la mayor o menor abundancia de agua (principal agente de la meteorización) y de la vegetación. Otro factor asociado es la temperatura y sus oscilaciones. Destaquemos, que cada aumento de 10°C de la temperatura duplica la velocidad a la que se producen la mayoría de las reacciones químicas.

La litología tiene una influencia decisiva sobre determinados mecanismos. Hay rocas, como las cuarcitas, que por su estabilidad química apenas son afectadas por los procesos de meteorización química, y por su dureza, tampoco por los de tipo físico; por eso, normalmente aparecen formando altos topográficos. Otras presentan distintas características en función del clima. Los granitos se alteran con gran facilidad en climas cálidos por la hidrólisis de sus feldespatos, mientras que en climas fríos y secos resisten bien los efectos de la meteorización. De igual manera, las calizas necesitan climas cálidos y húmedos para que se produzca su disolución. Una observación importante es que en las rocas ígneas la estabilidad de los minerales que las forman es contraria al orden en que se forman, definido por la denominada Serie de Bowen.

Factores asociados al litológico son la porosidad y permeabilidad que pueda presentar la roca, y su mayor o menor grado de fracturación tectónica, que favorecen la infiltración de aguas superficiales, favoreciendo a su vez los procesos de meteorización química y/o biológica.

La topografía, o las formas locales del relieve, pueden afectar a algunos de los mecanismos activos de erosión: por ejemplo, las laderas de solana sufren procesos distintos que los de las de umbría. En las primeras los veranos serán favorecedores de los procesos que implican la insolación, mientras que en las segundas durante los inviernos la acción del hielo podrá ser un agente erosivo importante. También el hecho de que exista una pendiente favorece procesos distintos a los propios de las planicies; en las primeras el agua discurre arrastrando los iones, mientras que en las segundas se produce un contacto más continuado entre el agua cargada de sales y las rocas. Así, por ejemplo, la laterización requiere un relieve muy suave.

La actividad biológica afecta también a los mecanismos de meteorización activos. En términos generales, la presencia de una cubierta vegetal continua favorece los procesos de meteorización química, mientras que la ausencia de ésta favorece los de tipo físico.

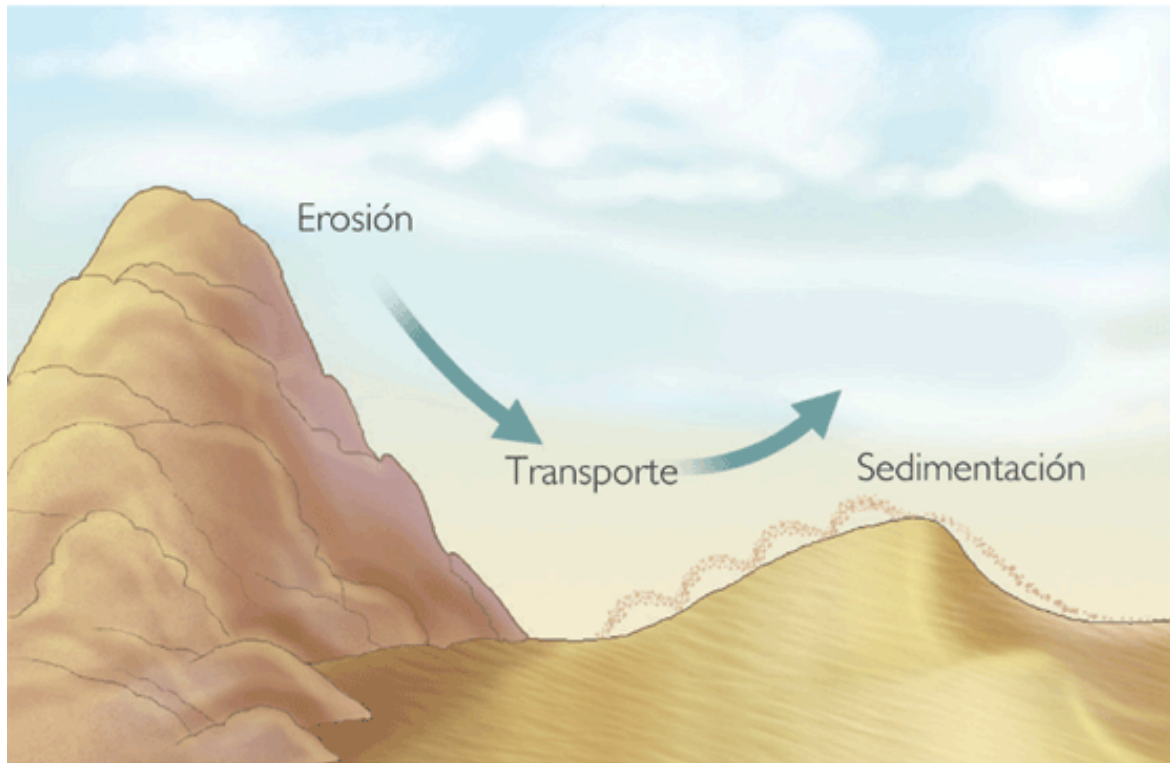


Figura 30: Ciclo de la Erosión (extraída de <http://www.madrimasd.org/blogs/universo/2008/08/15/98822>)

El tiempo favorece los procesos de meteorización, en general: todos estos procesos son de carácter lento, con lo que cuanto más tiempo queden sometidas las rocas a la acción de la intemperie, mayor facilidad tendrán los procesos erosivos para actuar. Así, si las rocas que albergan un depósito mineral son rápidamente cubiertas por otras, éste será preservado de los procesos erosivos. En este sentido, la tectónica regional puede jugar un importante papel.

Gracias a los datos almacenados, se puede hacer un estudio de estos factores para intentar minimizar sus efectos sobre las estructuras estudiadas, para ello pueden usarse varios métodos empíricos para calcular la erosión total laminar y en surcos en un segmento de pendiente. Uno de estos métodos es la Ecuación Universal de Pérdida de Suelos (USLE), es un método que utiliza seis factores: erosividad de la lluvia (R), susceptibilidad de erosión del suelo (K), largo de la pendiente (L), magnitud de la pendiente (S), cubierta y manejo de cultivos y residuos (C) y prácticas de conservación (P), para estimar la pérdida de suelos promedio (A) por el periodo de tiempo

representado por R, generalmente un año, con todo esto la fórmula final queda de la forma: $A = R \times K \times L \times S \times C \times P$.

Donde:

A = Es la pérdida de suelos calculada por unidad de superficie, expresada en las unidades seleccionadas para K y el período seleccionado para R, generalmente toneladas (t) hectárea (ha)⁻¹ año⁻¹.

R = El factor lluvia y escurrimiento, es el número de unidades de índice de erosión pluvial (EI), más un factor para escurrimiento por derretimiento de nieve o aplicación de agua. EL EI para una tormenta es el producto de la energía total de la tormenta (E) y su máxima intensidad en 30 minutos (I).

K = El factor susceptibilidad de erosión del suelo, es la tasa de pérdida de suelos por unidad EI para un suelo específico, medido en un porción de terreno estándar (22.13 m de largo, 9% pendiente, en barbecho y labranza continua).

L = El factor de largo de la pendiente, es la proporción de pérdida de suelos en el largo de la pendiente especificada con respecto a un largo de pendiente estándar (22.13 m).

S = El factor de magnitud de la pendiente, es la proporción de pérdida de suelos de una superficie con una pendiente específica con respecto a aquella en la pendiente estándar de 9%, con todos los otros factores idénticos.

C = El factor cubierta y manejo es la proporción de pérdida de suelo una superficie con cubierta y manejo específico con respecto a una superficie idéntica en barbecho, con labranza continua.

P = El factor de prácticas de apoyo de conservación, es la proporción de pérdida de suelo con un práctica de apoyo como cultivo con contorno, barreras vivas, o cultivo en terrazas, con respecto a aquella labranza en el sentido pendiente.

Otra ecuación que se puede usar es la Ecuación Universal Revisada de pérdida de Suelos (RUSLE), ésta ecuación fue desarrollada para superar algunas de las limitaciones de la anterior ecuación. Sus avances incluyen:

- Computarización de algoritmos para facilitar los cálculos.
- Nuevos valores de erosividad de lluvias-escurrimientos (R)
- Desarrollo de un término de susceptibilidad del suelo a la erosión estacionalmente variable (K) y métodos de estimación de K cuando el nomograma no es aplicable.

-Un nuevo método para calcular el factor cubierta-manejo (C), utilizando subfactores que incluyen uso previo de la tierra, cubierta de cultivos, cubierta vegetal del suelo y rugosidad del terreno.

-Nuevas formas de estimar los factores de largo y magnitud de la pendiente (LS) que consideran porcentajes de erosión en surcos e íter-surcos.

-La capacidad de ajustar el LS para pendientes de forma variable.

-Nuevos valores de prácticas de conservación (P) para cultivo en fajas alternadas, uso de drenaje subterráneo y praderas.

RUSLE aún está siendo desarrollado, y tal vez se introduzcan modificaciones.

Las fórmulas anteriores son muy sencilla, en caso de requerir más precisión se puede usar la de Rendimiento de Sedimento de una Pendiente Compleja la metodología AOF es un procedimiento matemático para estimar los potenciales de desprendimiento y transporte del suelo, incluyendo las proporciones relativas de erosión en surcos e íter-surcos, para laderas compuestas por uno o más segmentos con distintas características, la fórmula sería: $Y = W \times K \times L \times S \times C \times P$ donde:

Y = Es el rendimiento de sedimento en $t\ ha^{-1}$

W = Es un término de energía que es la suma de los componentes de energía pluvial y de escurrimiento. El componente de energía pluvial se deriva del índice de erosión pluvial (EI) de USLE, y el componente de energía de escurrimiento es una función de la tasa de escurrimiento de tempestad. Los otros factores son iguales que en el USLE.

Las fórmulas anteriores son para erosión del suelo por líquidos, como se ha visto anteriormente el viento también produce erosión y para calcularla podemos usar la fórmula de la Erosión Eólica $WE = f(I, WK, WC, WL, VE)$ donde:

I = Es el factor de susceptibilidad de erosión del suelo.

WK = Es el factor de escarpado de los camellones del suelo.

WC = Es un factor climático.

WL = Es el promedio de distancia descubierta recorrida por el viento a través del campo.

VE = Es la cubierta vegetal equivalente.

El modelo puede usarse para estimar el promedio anual de cantidad de erosión de un campo, o se puede especificar una cantidad aceptable de erosión y resolver la ecuación para determinar la cantidad de residuos, características de los camellones, y ancho de campo necesario para reducir la erosión a ese nivel.

Usando estas fórmulas se puede prever el desgaste de los suelos donde se encuentren las estructuras a estudiar, además al existir el modelo 3D se pueden hacer simulaciones de modificaciones del terreno para evitar la erosión colocando canales para el agua, evitando así que circule por lugares que no queremos estropear, también se pueden hacer simulaciones de cómo afectaría la colocación de paravientos en ciertas localizaciones para evitar la erosión eólica de la estructura, todo ello, nos permitiría conocer las localizaciones más idóneas y las estructuras más adecuadas para el fin que deseamos ayudando a reducir costes y un mejor mantenimiento.

Otro factor que afecta a las estructuras de los edificios históricos según el profesor de la UGR Eduardo Sebastián es la contaminación, la polución atmosférica, los gases emitidos por los vehículos en las ciudades, la humedad y los cambios de temperatura, son las principales causas de deterioro de la piedra y otros materiales de construcción, pero según éste profesor del departamento de mineralogía y petrología de la universidad de granada, la contaminación es el más determinante y el que más influye en el deterioro de los edificios históricos.

El grupo de investigación que dirige Eduardo Sebastián Pardo estudia los materiales de construcción de edificios históricos, no sólo la piedra ornamental sino los ladrillos y cerámicas “artísticas”, los morteros, hormigones, y han realizado el estudio sobre el estado de alteración y sus causas. Para grupos como éste disponer de un software como el presentado en este documento podría servirles para usando algoritmos de simulación, hacer estudios futuros de cómo afectará la contaminación a diversos elementos después de añadir la información a la base de datos. También permitiría estudiar posibles soluciones “virtualmente”, al disponer de los modelos en 3D también se puede apreciar visualmente como afectan estos elementos al posible deterioro de elementos históricos.

Restauración de estructuras:

Otra posible rama de ampliaciones para el proyecto podría ser la de restauración de estructuras, una ampliación para el proyecto podría ser la de análisis de posibilidades de cara a una restauración de una estructura, al estar dividido el modelo por partes, se pueden usar algoritmos que calculen las cargas y buscar los elementos que más carga soportan, ya que serán los primeros que haya que restaurar o reforzar.

Para conocer la presión producida por el viento se calcula con la fórmula:

$$P = C_p \cdot q \cdot S_4$$

Donde:

P = Presión estática

q = Velocidad convertida en presión dinámica.

Vs = Velocidad del viento en k.p.h (km/hora). Para determinar la velocidad, Vs, se cuenta con los mapas de amenaza eólica del país, donde por energía sabemos que la energía cinética es $\frac{1}{2} mV^2$ y m es la densidad del aire.

S₄ = Variación de la densidad del aire con la altura del nivel del mar.

C_p = Coeficiente de presión que depende de la forma de la edificación.

Para encontrar la presión ejercida sobre las diferentes partes de la estructura se emplean los coeficientes C_p que modifican el valor de la presión del viento básica para tener en cuenta los efectos de la forma de la edificación y el sentido de la presión que se produce

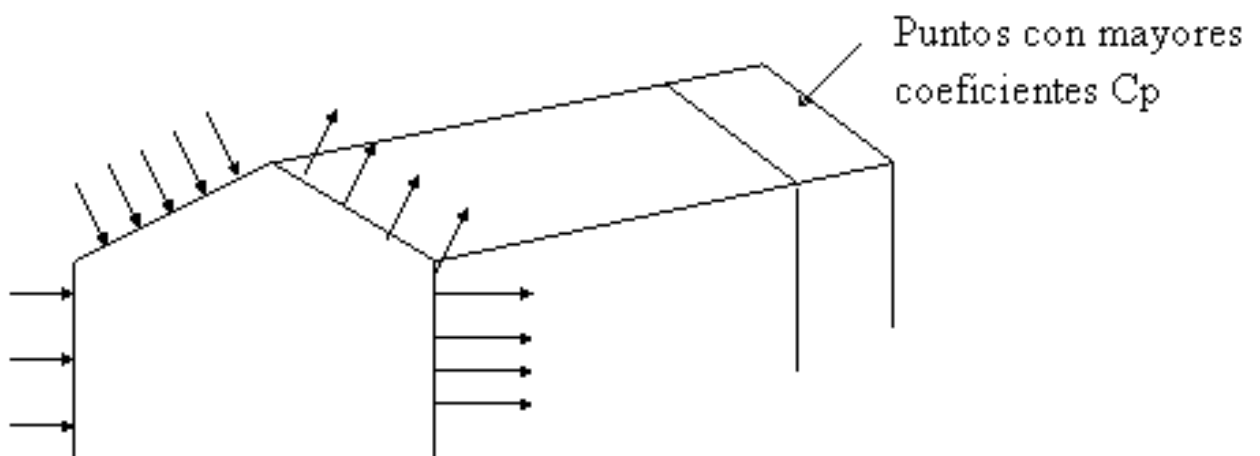


Figura 31: Puntos de fuerza (extraída de <http://estructuras.eia.edu.co/estructuras/cargas/fuerzas%20y%20metodos.htm>)

También se pueden crear modelos 3D de las ampliaciones o restauraciones para visualizar el impacto que podría tener en la estructura final y en el entorno permitiendo apreciar como quedarían las restauraciones y ampliaciones antes de empezar las obras ahorrando dinero y recursos, para ello habría que modelar las partes nuevas y acoplarlas a las escaneadas previamente que se encontrarían en la base de datos para después analizar el impacto que esta modificación o ampliación acarrearía en el contexto de la obra escaneada.



Figura 32: Teatro romano de Sagunto después de la restauración (extraída de <https://www.facebook.com/photo.php?fbid=489063704512238&set=a.241420309276580.59726.129134990505113&type=1&theater>)

Para casos en el que la restauración podría afectar en la estructura original y se deseara poder visualizar como era antiguamente, se podría recrear mediante modelados 3D una recreación de la estructura original, y gracias al visualizador web, se podría tener acceso a la recreación sin estropear los restos.

En los últimos años las instituciones públicas han puesto mucho empeño en la eficiencia energética, hasta el punto que hay que pasar unos controles de eficiencia a la hora de poder alquilar o comprar una casa. Observando estas nuevas medidas, muchas personas están haciendo reformas y tomando medidas en sus hogares para conseguir una mejor calificación energética, por ello en éste ámbito que tan en auge se encuentra podríamos usar nuestro producto creando unas pequeñas ampliaciones.

Para que éste proyecto se pueda emplear en este ámbito, se podrían desarrollar previamente los modelos de las estructuras a estudiar energéticamente, para después en la herramienta de visualización poder apreciar la energía utilizada por ésta.

Por ejemplo, se podría crear el modelo 3D de una vivienda a estudiar, localizar la posición a la que se encuentra y hacia donde está localizada, temperaturas medias en verano e invierno, etc. Con estos datos se podrían mirar de localizar ventanas para aprovechar mejor las horas de sol y que con menos coste energético de los elementos de climatización de la vivienda se pueda mantener una temperatura idónea, también se podría controlar la posición de éstos elementos de climatización para que con el mínimo coste mantuvieran mejor la temperatura, estudiar la posibilidad de poner placas solares para calentar el de la casa, etc.

Bibliografía

- A 3D web GIS system Base don VRML and X3D de Wang Ming
- http://es.wikipedia.org/wiki/Sistema_de_Información_Geográfica
- <http://www.monografias.com/trabajos14/informageogra/informageogra.shtml>
- http://geomaticblog.net/2007/11/15/2007-11-15_mas_reflexiones_acerca_del_sig_3d_grandes_escalas/
- <http://mayaarch3d.unm.edu/index.php>
- <http://ieeexplore.ieee.org/Xplore/home.jsp>
- <http://www.monografias.com/trabajos14/informageogra/informageogra.shtml>
- <http://www.ign.es/ign/layoutIn/actividadesSistemaInfoGeografica.do>
- <http://www.oas.org/dsd/publications/Unit/oea65s/ch10.htm>
- <http://www.ambientum.com/revista/2010/febrero/aplicaciones-medioambientales-SIG.asp>
- <http://forest.jrc.ec.europa.eu/effis/applications/fire-history/>
- <http://www.uclm.es/users/higueras/yymm/YM4.html>
- <http://www.fao.org/docrep/t2351s/t2351s03.htm>
- <http://noticias.universia.es/ciencia-nn-tt/noticia/2002/03/26/636677/contaminacion-es-primer-mal-afecta-piedra-edificios-historicos-profesor-ugr-eduardo-sebastian.html>
- <http://estructuras.eia.edu.co/estructurasI/cargas/fuerzas%20y%20metodos.htm>
- [http://es.wikipedia.org/wiki/Unity_\(software\)](http://es.wikipedia.org/wiki/Unity_(software))
- http://es.wikipedia.org/wiki/Microsoft_Visual_Studio
- http://es.wikipedia.org/wiki/C_Sharp
- <http://es.wikipedia.org/wiki/AJAX>
- <http://es.wikipedia.org/wiki/JavaScript>
- http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada
- <http://es.wikipedia.org/wiki/PhpMyAdmin>
- <http://es.wikipedia.org/wiki/HTML>