

TESIS DOCTORAL

Una Contribución a la Modelización y Virtualización Numérica de Estructuras Arquitectónicas

**(Una aplicación práctica a la estructura del Sagrario de la Catedral
de la Ciudad de México)**

**Tesis presentada por:
Arq. Francisco Muñoz Salinas**

Para obtener el grado de: Doctor Arquitecto

Director de Tesis: Javier López-Rey Laurens

Codirector: Jordi Maristany i Carreras

**Programa de Doctorado: Análisis de Estructuras Arquitectónicas
Departamento de Estructuras en la Arquitectura
Escuela Técnica Superior de Arquitectura de Barcelona
Universidad Politécnica de Cataluña**

Barcelona 2000

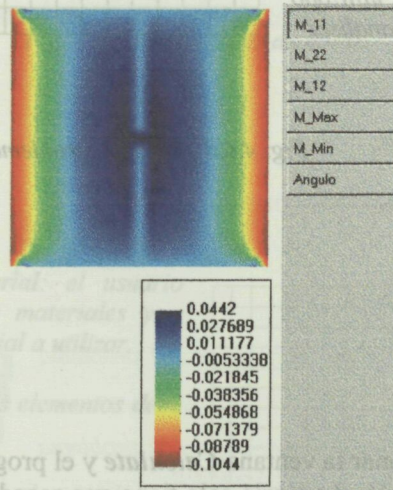
4.6.3 Postproceso

Como puede apreciarse, todo el *preproceso* realizado hasta el momento ha sido de forma sencilla y dinámica. Esto, muestra la eficacia del programa *GiD* en la fase de la preparación de los datos y la generación de malla, previa al análisis de la estructura. Este análisis, es realizado con programas informáticos, ya sean comerciales o creados por el usuario.

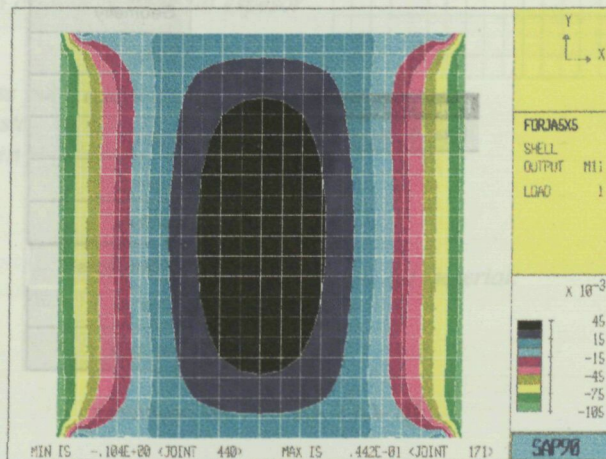
Finalmente, mostraremos los resultados obtenidos en el análisis y la facilidad con que se pueden configurar en el *postproceso* y manipular los datos generados de manera gráfica.

Al igual que en el capítulo 9, se presentaran los resultados del análisis en los dos formatos, el *postproceso* de *GiD* y el de *Sap90*.

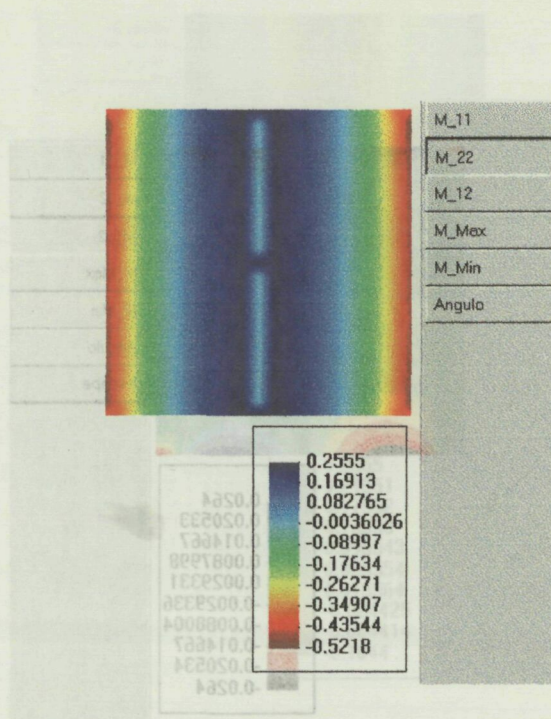
* ESFUERZOS OBTENIDOS



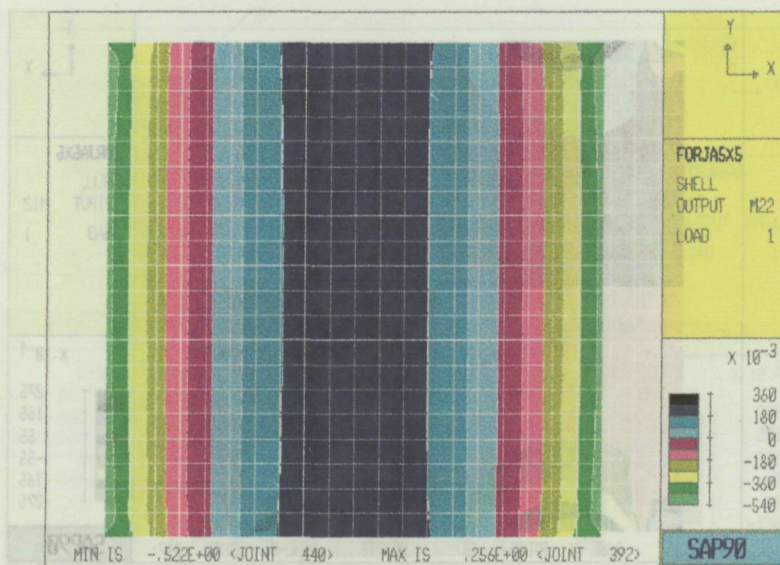
Momento 11 del postproceso de GiD



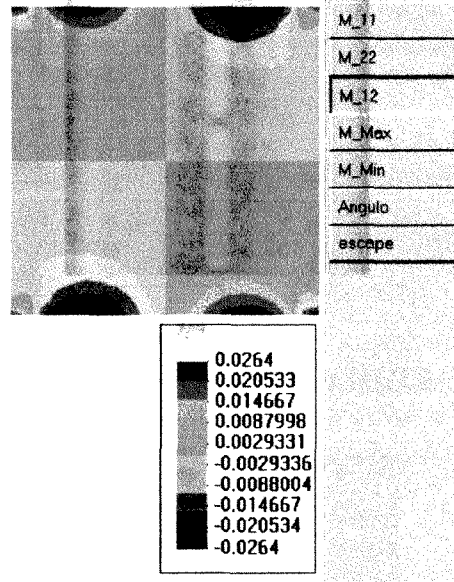
Momento 11 del postproceso de Sap90



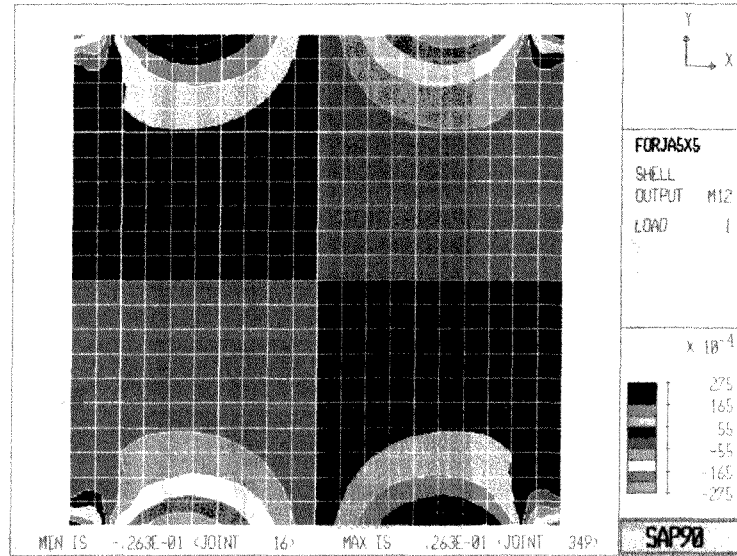
Momento 22 del postproceso de GiD



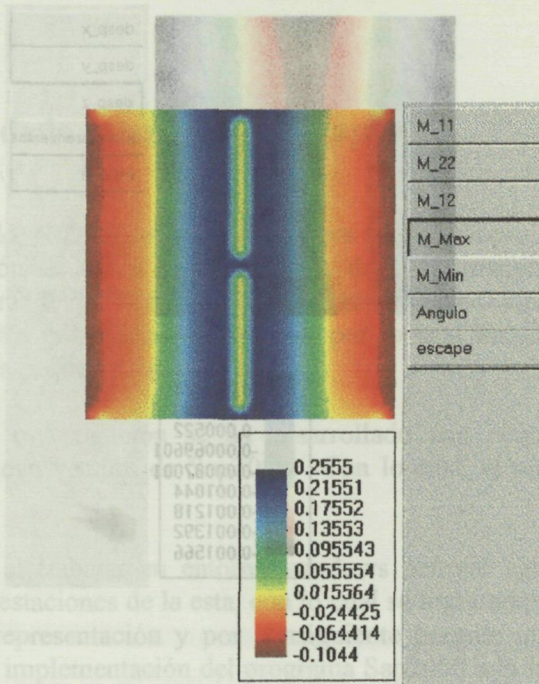
Momento 22 del postproceso de Sap90



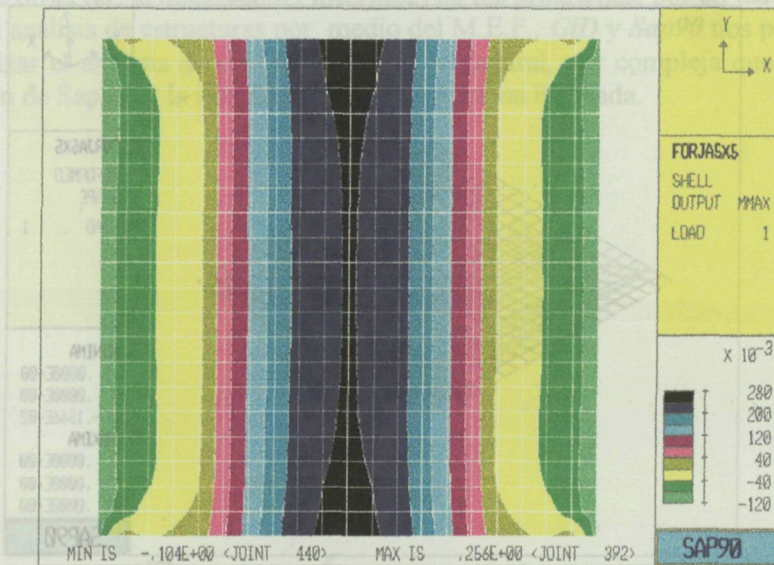
Momento 12 del postproceso de GiD



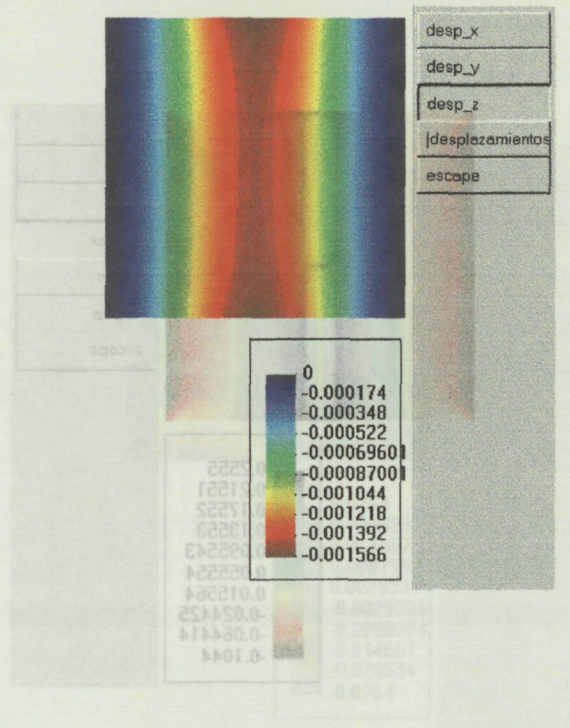
Momento 12 del postproceso de Sap90



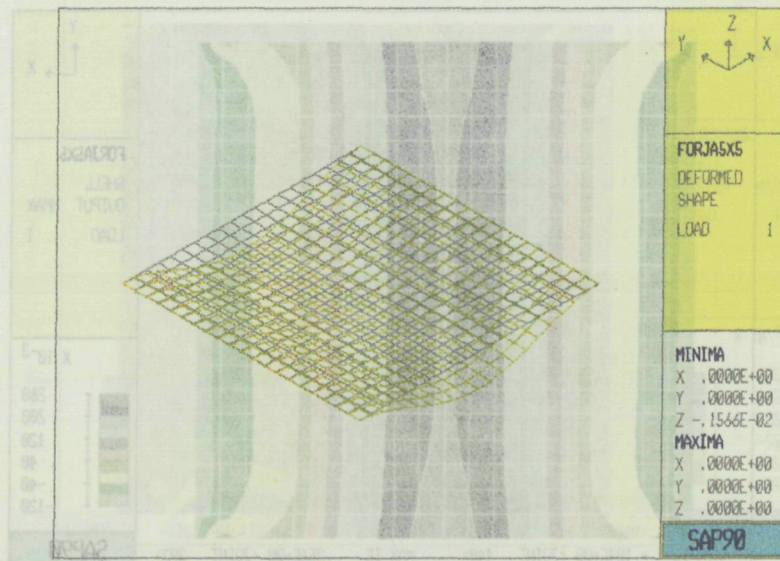
GiD Momento Máximo del postproceso de GiD



Momento Máximo del postproceso de Sap90



Valores de Deformación del postproceso de GiD



Valores de Deformación del postproceso de Sap90

4.6.4 Limitaciones de la interface GiD-Sap90

Como ya se ha comentado, *GiD* es un programa de pre y postproceso sin limitación en la generación de mallas o la asignación de datos o condiciones; de igual manera permite visualizar cualquier resultado numérico por grande que sea el modelo. Sin embargo, el *Sap90* solo permite realizar análisis con un máximo de catorce mil grados de libertad; por lo cual los cálculos se limitan a elementos estructurales puntuales ó a pequeñas estructuras.

Por esta razón, en esta investigación se ha desarrollado una pequeña interface que permite la implementación de la nueva versión de *Sap2000*. Con lo cual, el análisis de estructuras se vuelve ilimitado.

Además, este programa al trabajar en entorno windows permite agilizar la interface con *GiD* de manera que mejora las prestaciones de la esta, con lo cual se logra ampliar los tipos de salida de datos gráficos para su mejor representación y por consiguiente permite una mejor interpretación de los resultados numéricos. La implementación del programa *Sap2000* a la interface original se describe en el apartado siguiente.

4.6.5 Conclusiones

Como se puede observar, la integración (*interface*) de los programas, por un lado el pre y postproceso y por el otro el análisis de estructuras por medio del M.E.F., *GiD* y *Sap90* nos permiten de forma ágil y sencilla realizar el análisis de cualquier género estructural, por compleja que sea. Además, con la implementación de *Sap2000* la interface se amplía de forma ilimitada.

5 Desarrollo de la Interface GiD-Sap2000-VRML

5.1 Nuevas interfaces gráficas para la virtualización numérica de edificios históricos (Interface *GiD-Sap2000*, *Sap2000-VRML*)

5.1.1 Introducción

Dado que el programa *Sap90* tiene limitaciones con los grados de libertad, se a optado por ampliar la interface *GiD-Sap90* implementado el análisis numérico de *Sap2000*. Esto permitirá al usuario estudiar el comportamiento estructural de edificaciones más grandes y más complejas.

La ampliación de esta interface se realizó utilizando la misma metodología del proceso de *GiD* con *Sap90*, ya que el archivo que genera *GiD*, dentro del preproceso, se puede exportar a *Sap2000*. Por tanto, toda vez que se ha exportado el archivo, *Sap2000* lo convierte a su propio formato con lo cual es posible iniciar el análisis dentro de su entorno.

5.1.2 Interface *GiD-Sap2000*

El proceso de la interface de *GiD* con *Sap2000* es bastante sencillo, ya que solo hay que preparar la estructura dentro del preproceso utilizando cualquiera de las interfaces de *GiD-Sap90*; ya sea con *Shell*, *Shell3d*, *Asolid* ó *Solid*.

Una vez que se ha realizado este proceso, es necesario solicitarle a *GiD* que genere el archivo para su cálculo.

Este archivo debe de llevarse a *Sap2000* e importarlo. *Sap2000* traduce y prepara el archivo en su propio formato y avisa si se esta en condiciones para realizar el análisis. Es entonces, cuando se inicia el cálculo de la estructura.

Esto es posible, gracias a la interface desarrollada para el preproceso de *GiD* con *sap2000*.

Metodología desarrollada para la interface *GiD-Sap2000*

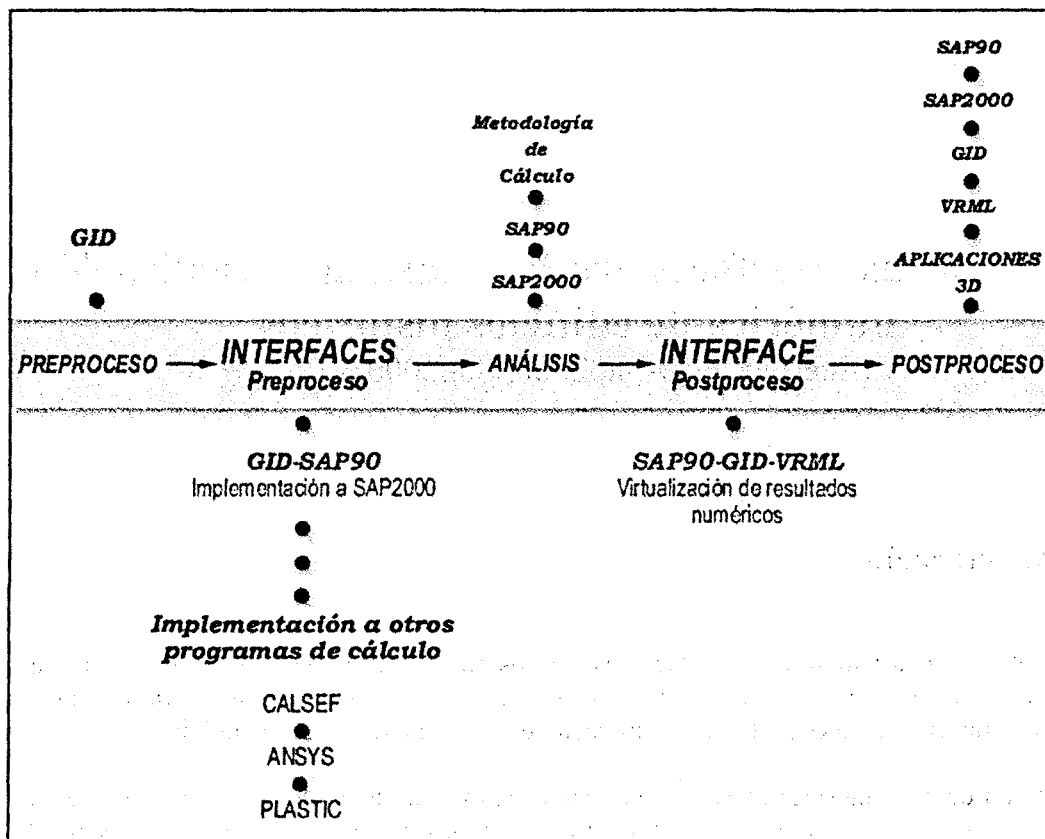


Fig. 5.1.1 Proceso para la implementación de Sap2000 en la interface GiD-Sap90

Al igual que en el postproceso de la interface *Sap90-GiD*, aquí se han generados una serie de programas. Los cuales, traducen toda la información generada por Sap2000 convirtiéndola en formato *GiD*. Los programas fuentes se encuentran en los anexos de la misma interface.

La información que Sap2000 genera, corresponde a todos los esfuerzos y tensiones y los valores numéricos los asigna a los nodos. Estos valores, gracias a la interface 3dshell, son tomados y procesados generando una interpolación de todos los valores que correspondan al mismo nodo, así como al mismo nombre ya sea de esfuerzos o de tensiones.

La interface limpia, procesa y edita todos los datos, generados por Sap2000, en formato *GiD* para el postproceso. En resumen, para desarrollar la interface *GiD-Sap2000* se implemento la misma metodología desarrollada para la interface *GiD-Sap90* y solo cambian los programas que transforman la información.

5.1.3 Ventajas de la interface *GiD-Sap2000*

Mientras que con la interface con *sap90*, el análisis no podía rebasar las quince mil ecuaciones (Alrededor de dos mil cuatrocientos grados de libertad), ahora con la interface con *Sap2000* el análisis se vuelve ilimitado; convirtiéndose, por tanto, en una gran ventaja dentro del análisis.

El programa *Sap2000*, a diferencia del 90 trabaja dentro del entorno windows. Por tanto, su utilización se vuelve más dinámica y de sencilla utilización.

En lo que corresponde a *GiD*, gracias a que el análisis se realiza mediante *Sap2000*, su postproceso se vuelve mucho más interesante ya que es posible visualizar y analizar modelos más grandes sin problemas de limitaciones. Esto, proporciona una herramienta muy potente dentro del análisis de estructuras arquitectónica y en particular, de edificios históricos.

Hay que recordar, que dentro del estudio del comportamiento estructural de edificios histórico el problema fundamental, debido a su geometría compleja, es el de generar un modelo numérico adecuado. Por todo ello, el uso de estas interfaces es fundamental; ya que reduce en gran medida el tiempo utilizado en el estudio de este género de edificios.

Además, con la implementación de la interface *GiD-Sap2000* el usuario tiene más alternativas de visualización de los resultados numéricos. Con esta investigación, se abre la oportunidad de poder escoger, de entre varias opciones, la forma en que cada persona desee examinar los resultados gráficos obtenidos.

Esto quiere decir, que la persona que utilice esta interface tiene la opción de decidir si desea visualizar los resultados dentro del entorno de *Sap2000* o en su defecto irse a *GiD* y continuar con análisis. Por ello, a continuación se resumen algunas ventajas que cada programa ofrece gracias a este trabajo de investigación.

Opciones que ofrece *sap2000* en el preproceso:

- Admite e importa los archivos generados en formato *Sap90* para su análisis.
- Su preproceso, contiene varios modelos básicos; los cuales se pueden modificar y adaptar para diferentes tipologías de análisis.
- Permite la importación de archivos de dibujo en formato *dxf*.
- Tiene la opción de generar malla, pero al igual que en el *sap90* su generación es lineal y ortogonal.
- Sigue trabajando con elementos cuadrilaterales.
- Gracias a que trabaja en entorno windows, tiene la posibilidad de desarrollar todo el preproceso de una estructura (condiciones de borde, características de los materiales, datos generales) por medio de ventanas e iconos propios del programa.
- Además, al igual que *sap90*, se puede generar un archivo de texto con todos los datos del análisis, pero con su nuevo formato.

Dentro del cálculo, sap2000 implementa análisis en segundo orden, además proporciona los resultados de esfuerzos y tensiones; así como los de Von Mises, cosa que sap90 no desarrollaba en el mismo análisis.

Opciones que ofrece sap2000 en el postproceso:

- La visualización gráfica de los resultados, gracias a windows, es más dinámica.
- Crea un archivo de animación de deformadas, así como del los modos de vibración en un análisis dinámico.
- Permite observar los resultados de la estructura desde diferentes puntos de vistas, pero siempre desde fuera de ella.
- Tiene herramientas dinámicas de visualización, tales como: zoom por ventana ó por pasos, restauración de la vista original, pan (desplazamientos dinámicos por pantalla a lo largo de la estructura)
- Realiza animaciones por pantalla de la deformada, con la opción de incluir los resultados gráficos.
- Con lo que respecta a la salida de información, permite imprimir los resultados numéricos en archivo de texto o gráficos en papel.

En resumen, estas son algunas de las opciones más importante que Sap2000 implementa en su programa.

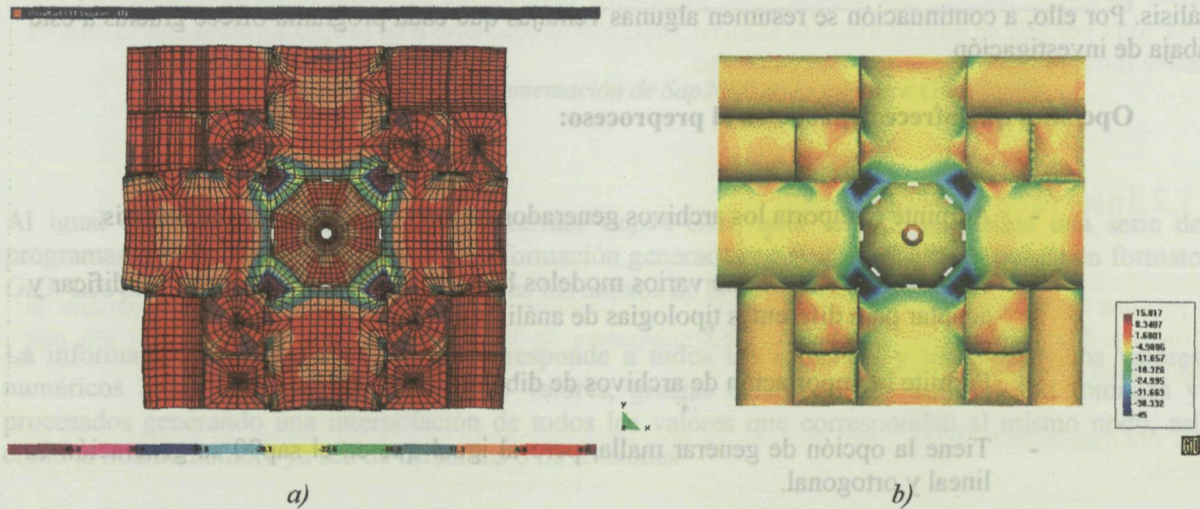


Fig. 5.1.2 Opciones de visualización
a) Entorno Sap2000 b) Interface GiD-Sap2000

Con lo que respecta a *GiD*, aquí, no se definen las opciones que tiene *GiD* en su pre y postproceso ya que en capítulos anteriores se han detallado de manera más explícita.

5.1.4 Comparación de los resultados gráficos de la interface GiD-Sap2000

El desarrollo de esta interace, cumplimentando a la de sap90, aparece como una nueva herramienta para el análisis de edificios arquitectónicos. Herramienta que puede ser utilizada tanto para investigadores, calculistas; así como en la enseñanza sobre el comportamiento estructural de las estructuras en general.

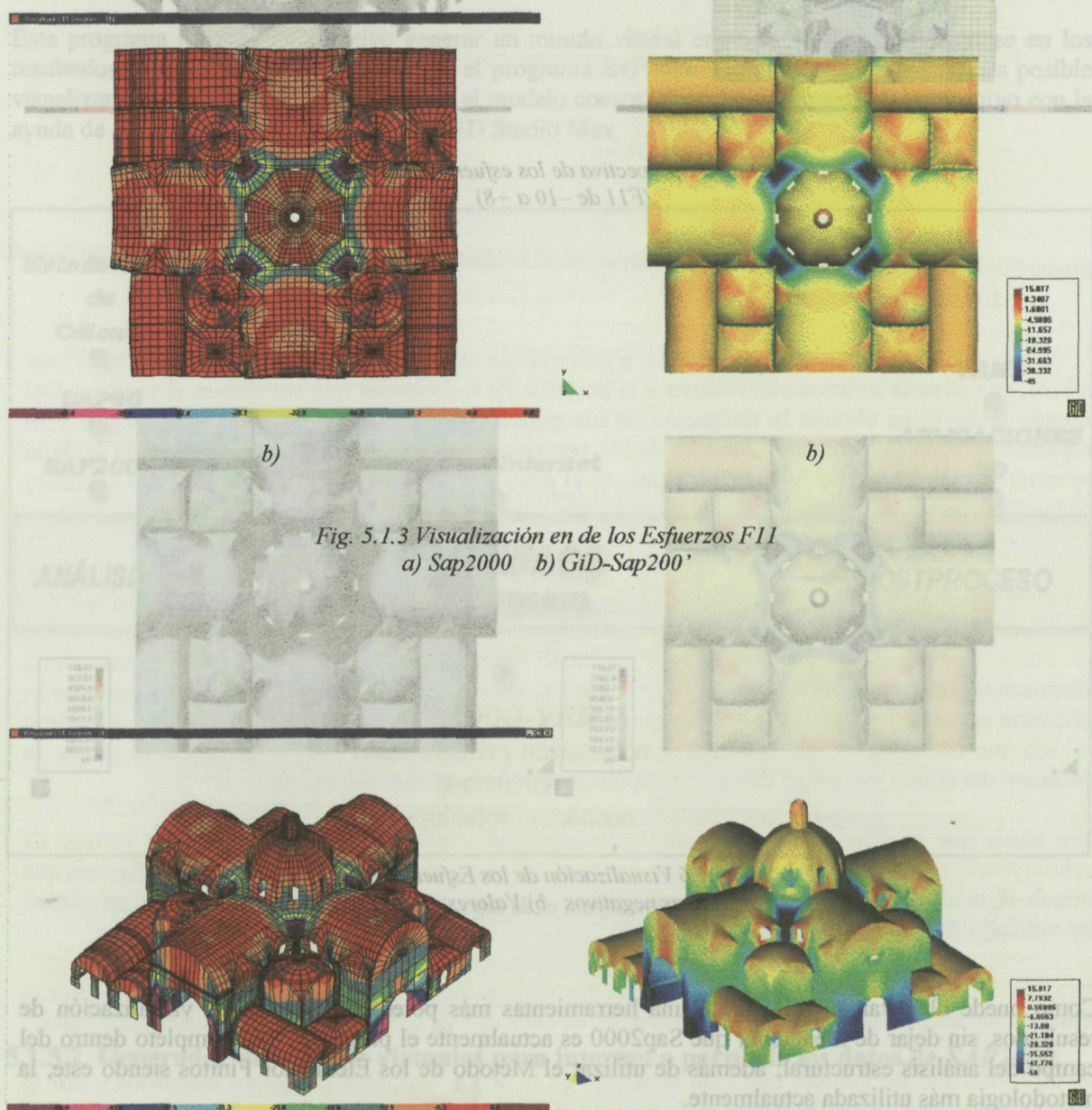


Fig. 5.1.3 Visualización en de los Esfuerzos F11
a) Sap2000 b) GiD-Sap2000'

Fig. 5.1.4 Perspectivas en Sap2000 y la inteface GiD-Sap2000

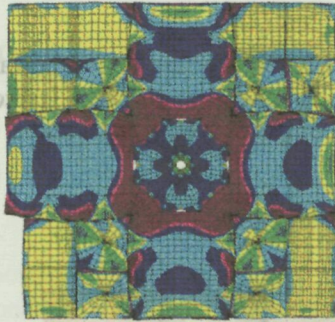


Fig. 5.1.5 Planta y perspectiva de los esfuerzos F11 en Sap2000
(F11 de -10 a +8)

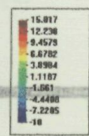
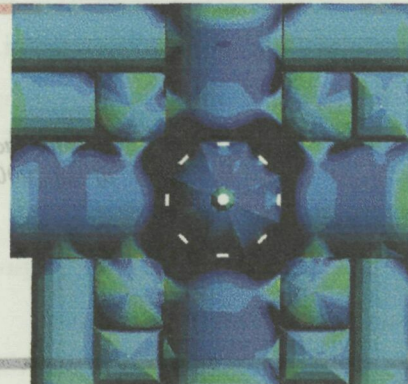
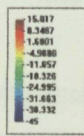
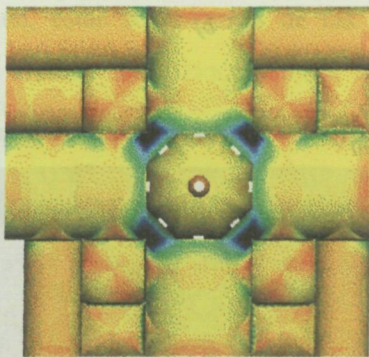


Fig. 5.1.6 Visualización de los Esfuerzos F11
a) Valores negativos b) Valores positivos

Como puede observarse, *GiD* ofrece una herramienta más potente dentro de la visualización de resultados, sin dejar de mencionar que *Sap2000* es actualmente el programa más completo dentro del campo del análisis estructural; además de utilizar el Método de los Elementos Finitos siendo este, la metodología más utilizada actualmente.

5.1.5 Interface *Sap2000-VRML*

5.1.5.1 Objetivos

Este programa tiene como objetivo generar un mundo virtual en formato VRML basándose en los resultados numéricos proporcionados por el programa *SAP2000*. Con esta transformación es posible visualizar y hacer un recorrido virtual por el modelo construido, así como manipular el archivo con la ayuda de otros programas, por ejemplo el 3D Studio Max.

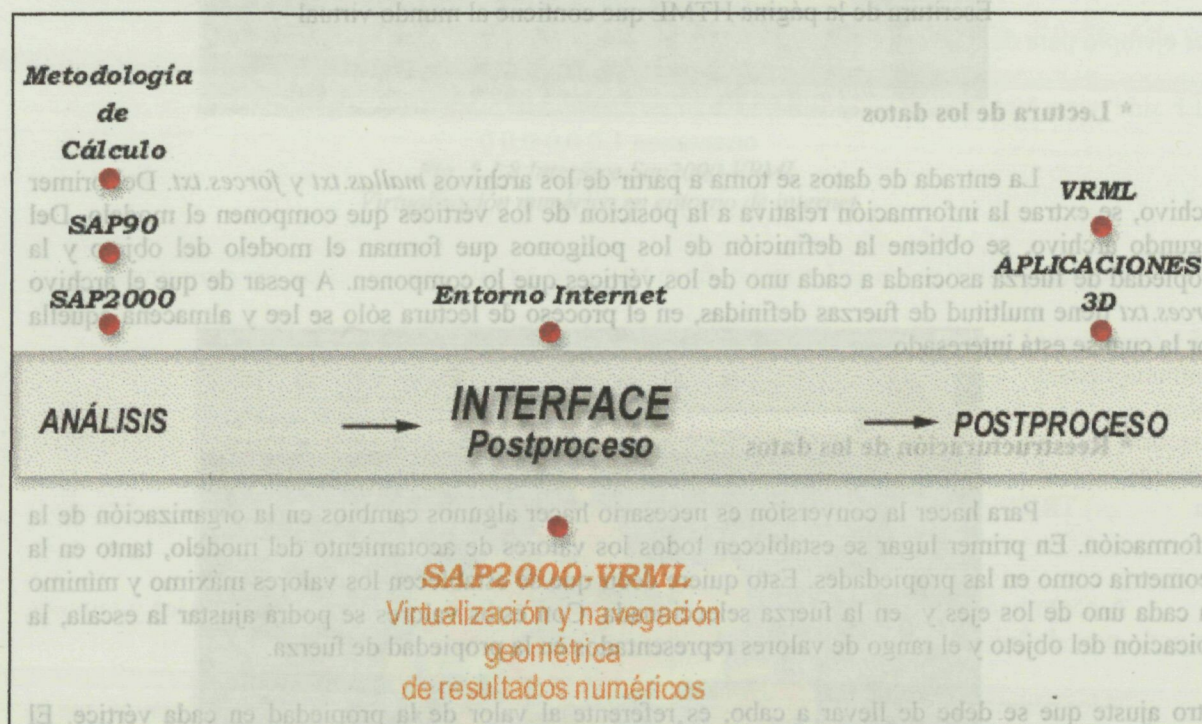


Fig. 5.1.7 Metodología de la interfaz *Sap2000-VRML*

5.1.5.2 Generador de mundos virtuales para internet a partir de los datos de *SAP2000*.

El formato VRML (Virtual Reality Modelling Language) es uno de los estándares actuales para la distribución de modelos tridimensionales que no sólo refleja la geometría del modelo, sino que también permite asignar propiedades y comportamientos tanto en el modelo como en la forma de visualizarlo y navegar dentro de él. En este caso en particular, las propiedades que se desean reflejar

en el modelo virtual corresponden a los valores numéricos obtenidos de cualquier análisis generado por el programa SAP2000.

La interface del programa fue construida en el entorno de desarrollo de Visual Basic, por la sencillez y rapidez que caracterizan a este entorno.

5.1.5.3 Metodología

El proceso de conversión pasa por las siguientes etapas:

- Lectura de los datos
- Reestructuración de los datos
- Escritura del modelo VRML
- Escritura de la página HTML que contiene al mundo virtual

* Lectura de los datos

La entrada de datos se toma a partir de los archivos *mallas.txt* y *forces.txt*. Del primer archivo, se extrae la información relativa a la posición de los vértices que componen el modelo. Del segundo archivo, se obtiene la definición de los polígonos que forman el modelo del objeto y la propiedad de fuerza asociada a cada uno de los vértices que lo componen. A pesar de que el archivo *forces.txt* tiene multitud de fuerzas definidas, en el proceso de lectura sólo se lee y almacena aquella por la cual se está interesado.

* Reestructuración de los datos

Para hacer la conversión es necesario hacer algunos cambios en la organización de la información. En primer lugar se establecen todos los valores de acotamiento del modelo, tanto en la geometría como en las propiedades. Esto quiere decir que se establecen los valores máximo y mínimo en cada uno de los ejes y en la fuerza seleccionada. Con estos valores se podrá ajustar la escala, la ubicación del objeto y el rango de valores representados en la propiedad de fuerza.

Otro ajuste que se debe de llevar a cabo, es referente al valor de la propiedad en cada vértice. El archivo *forces.txt*, define estas propiedades de forma independiente para cada uno de los polígonos que forman el modelo, lo que lleva a que un mismo vértice pueda tener distintos valor de propiedad dependiendo del polígono al cual esté asociado.

Para disminuir las discontinuidades producidas por este hecho, se ha decidido asociar a cada vértice el promedio de las propiedades que tiene asociadas en cada uno de los polígonos a los cuales pertenezca.

De tal forma que los vértices que sean interiores en la malla, tendrán 4 polígonos asociados y por lo tanto su valor final será el promedio de la propiedad en cada uno de ellos. En cambio un vértice que se encuentre en una esquina de la malla, sólo tendrá un polígono asociado y por lo tanto no hará falta hacer el promedio.

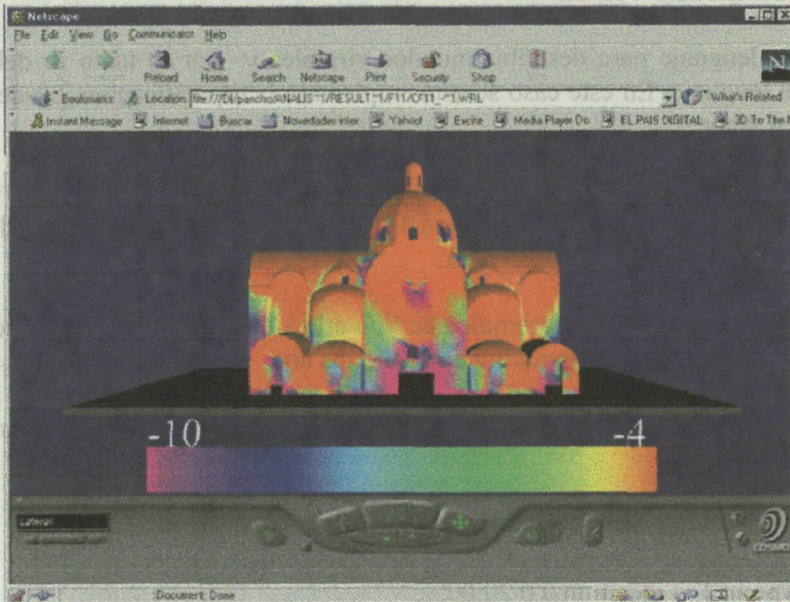


Fig. 5.1.8 Interface *Sap2000-VRML*
Virtualización numérica en entorno de internet

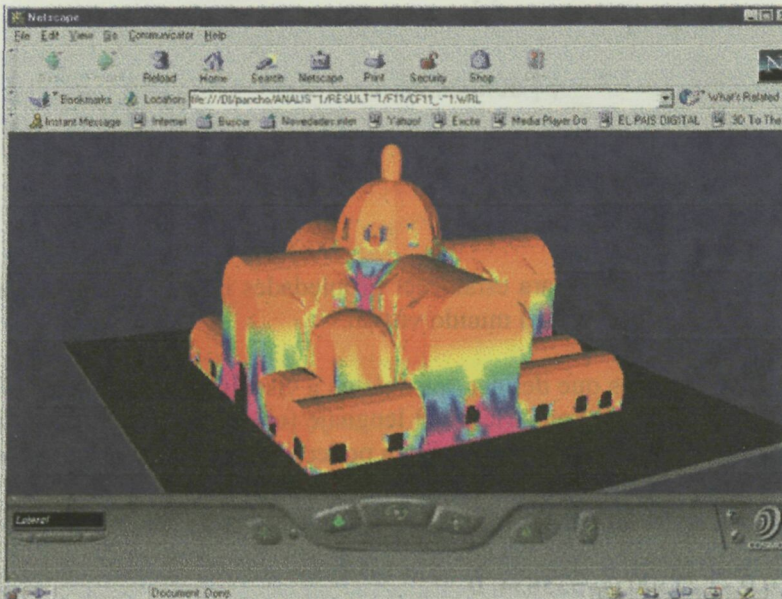


Fig. 5.1.9 Mundo virtual implementando los resultados numéricos de *Sap2000*

* Escritura del modelo VRML

VRML, es un lenguaje para describir mundos virtuales y por lo tanto se debe ajustar a la sintaxis que define el lenguaje. En este caso se está empleando la especificación 2.0 del lenguaje que es la especificación en vigor.

El mundo que se va a construir consta de varios objetos: la malla del modelo aportado por *SAP2000*, un plano de referencia sobre el cual asentar el modelo, las luces existentes en el mundo virtual y las especificaciones de puntos de vista del observador.

Un archivo VRML tiene una estructura jerárquica de tipo árbol. Por lo que, el primer nodo contiene a los demás y todo lo que afecte a éste repercutirá en los nodos hijo.

En este caso a partir de un nodo de grupo se añaden los demás nodos que representarán a los objetos mencionados anteriormente.

Por ejemplo para definir un nodo para un punto de observación.

```
DEF side_view Viewpoint {
    position 0 0 20.00
    orientation 1.0 0.0 0.0 0
    fieldOfView 0.387763
    description "Lateral"
}
```

Después seguirían los nodos que definen las luces, en este caso se utiliza luz direccional.

```
DEF front_light DirectionalLight {
    ambientIntensity .1
    color 1 1 1
    direction 0 0 -1
    intensity .5
    on TRUE
}
```

A continuación viene un nodo para establecer propiedades iniciales del navegador y también un nodo para definir un color de fondo en el mundo virtual.

Y los últimos nodos son los que definen los objetos a visualizar, el plano de referencia que se construye con una de las primitivas geométricas del lenguaje (un box) el cual está definido como un nodo hijo de un nodo transformación. Esto se hace para poder colocar el plano de referencia en el sitio adecuado dependiendo del modelo a visualizar.

```
Transform {
    translation 0.0 -2.21300549335834 0
    children [
        Shape {
            appearance Appearance {
                material Material { diffuseColor 0.2 0.2 0.2 }
            }
            geometry Box { size 8 0.01 8 }
        }
    ]
}
```

El nodo que representa la malla del modelo es bastante largo y por lo tanto sólo se mostrarán las partes que lo componen. Al igual que con el objeto del plano de referencia este objeto tendrá un nodo transformación que controla su posición y orientación y dentro de ese nodo se crea un nodo formado por polígonos (*IndexedFaceSet*).

```

Transform {
  scale 0.125997 0.125997 0.125997
  children [
    Transform {
      rotation 1 0 0 1.57
      translation -58.3524017333984 -10.2425394058228 41.4154014587402
      children
      Group {
        children
        Group {
          children
          Shape { # Superficie
            appearance Appearance {
              material DEF estrucMat Material { shininess 0.2 }
            }
            geometry IndexedFaceSet {
              coord Coordinate {point [ 0 0 0 ,
                34.5423 -18.7154 7.32142 ,
                .....
                ] }
              color Color { color [ 0.96 0.0 .96 ,
                ...
                0.35 0.06 ] }
              colorPerVertex TRUE
              coordIndex [
                5400 5290 5319 5499 -1
                ....
                ]
              colorIndex [
                2 0 0 0 -1
                ....
                ]
              ccw FALSE #tenia valor true
              solid FALSE
              convex FALSE
              creaseAngle 0
            }
          }
        }
      }
    ]
  }
}

```

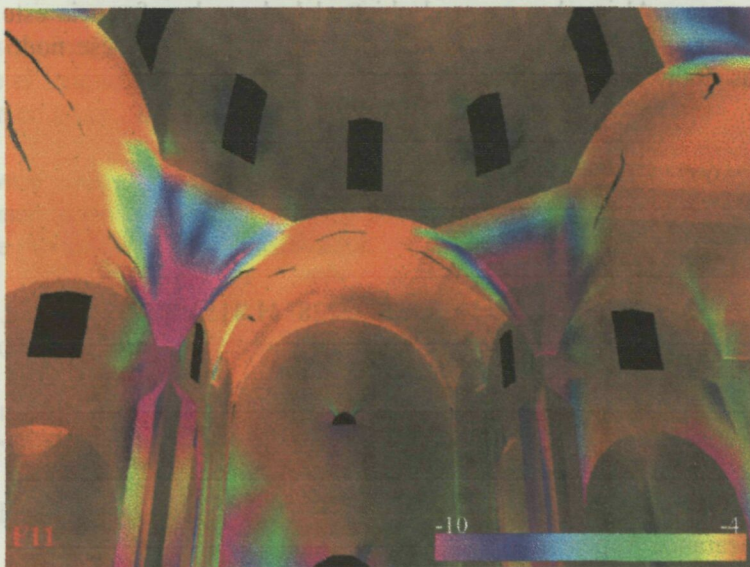



Fig. 5.1.10 Posibilidades de navegación virtual dentro del edificio.

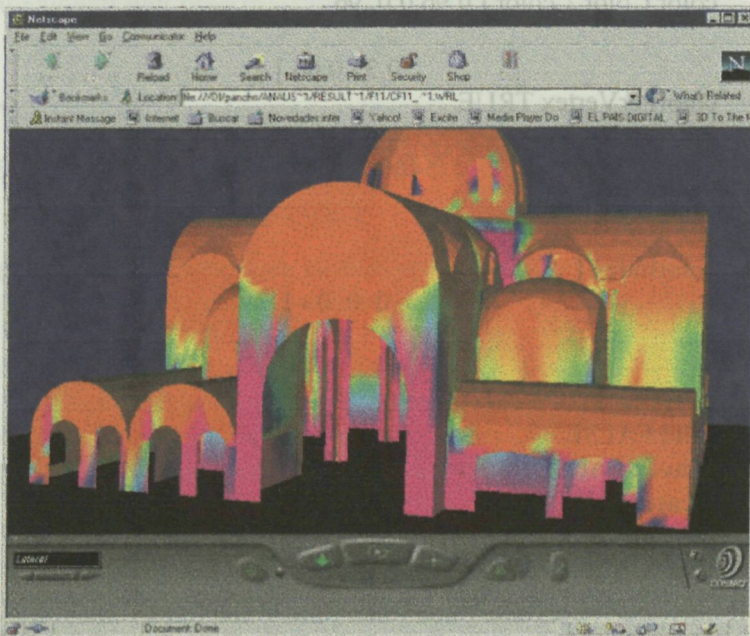


Fig. 5.1.11 Visualización virtual de los axiles

* Escritura de la página HTML que contiene al mundo virtual

Además del archivo VRML, se adjunta un archivo HTML, es decir, una página web desde donde se llama el archivo VRML. Esto es un paso que no es necesario para la visualización, ya que los navegadores de internet pueden leer el archivo VRML sin necesidad de esta página. Sin embargo, En la página web se anexa la escala de valores que se está empleando y se podría incluir más información referente al modelo. Un ejemplo de esta página es el siguiente.

```
<html>
<head>
<title>Modelo generado para la fuerza F22</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#CCCCCCFF">
<table width="100%" border="0" cellspacing="10">
<tr>
<td width="80" valign="top">
<p align="center"><font face="Arial, Helvetica, sans-serif"><b><font size="2"> 1.6884
</font></b></font></p>
<table width="20" border="0" cellspacing="0" align="center">
<tr bgcolor="#FF3300"><td>&nbsp;</td></tr>
<tr bgcolor="#FF9900"><td>&nbsp;</td></tr>
<tr bgcolor="#FFFF00"><td>&nbsp;</td></tr>
<tr bgcolor="#66FF00"><td>&nbsp;</td></tr>
<tr bgcolor="#33FF66"><td>&nbsp;</td></tr>
<tr bgcolor="#33FFCC"><td>&nbsp;</td></tr>
<tr bgcolor="#33CCFF"><td>&nbsp;</td></tr>
<tr bgcolor="#0066FF"><td>&nbsp;</td></tr>
<tr bgcolor="#6600FF"><td>&nbsp;</td></tr>
<tr bgcolor="#CC00FF"><td>&nbsp;</td></tr>
<tr bgcolor="#FF00CC"><td>&nbsp;</td></tr>
</table>
<p align="center"><font face="Arial, Helvetica, sans-serif" size="2"> <b>-1.4577</b></font></p>
</td>
<td><embed src="fvF22nuevo.wrl" width="800" height="600">
</embed></td>
</tr>
</table>
</body>
</html>
```


5.1.6 Implementación del programa 3D Studio Max para generar recorridos virtuales

3D Studio Max, es un programa gráfico el cual, transforma una geometría asignando a esta diversas texturas; ya sean en colores básicos o como de materiales.

Permite generar renders con esta texturas, además de recorridos, los cuales los convierte en formatos de vídeo. Importar geometrías en 3d en diferentes formatos gráficos. Uno de ellos es el VRML.

Para esta investigación, se decidió que la implementación del 3Dmax sería una más de las herramientas de visualización gráfica para el conocimientos del comportamiento estructural de los edificios; ya que permite importar los resultados obtenidos de la interface *Sap2000-VRML*.

A partir de esta implementación, el usuario puede decidir y generar recorridos puntuales, que le interesen, dentro de la estructura. Con ello, se logra visualizar cada espacio del edificios.

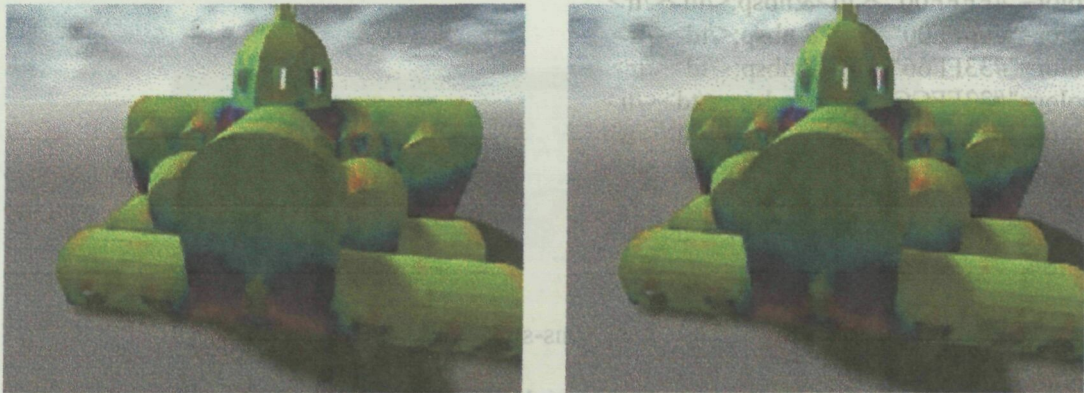


Fig. 5.1.12 Renderización de los resultados numéricos en 3Dmax

Como ya se ha comentado, a partir de archivo *vrmf* generado con la interface *Sap2000-VRML* el usuario puede entrar en el programa 3Dmax y comenzar a definir texturas del entorno que desee.

Una vez que se han definido todas las texturas externas al modelo, el usuario puede definir recorridos puntuales en función de las zonas claves que desee visualizar.

También es posible definir diferentes tipos y direcciones de la iluminación que se desee asignar al modelo

En el programa 3DMax se pueden generar renders de todos los frames definidos durante el recorrido, de la misma manera se pueden editar videos



Fig. 5.1.13 Virtualización de interiores y exteriores del Sagrario

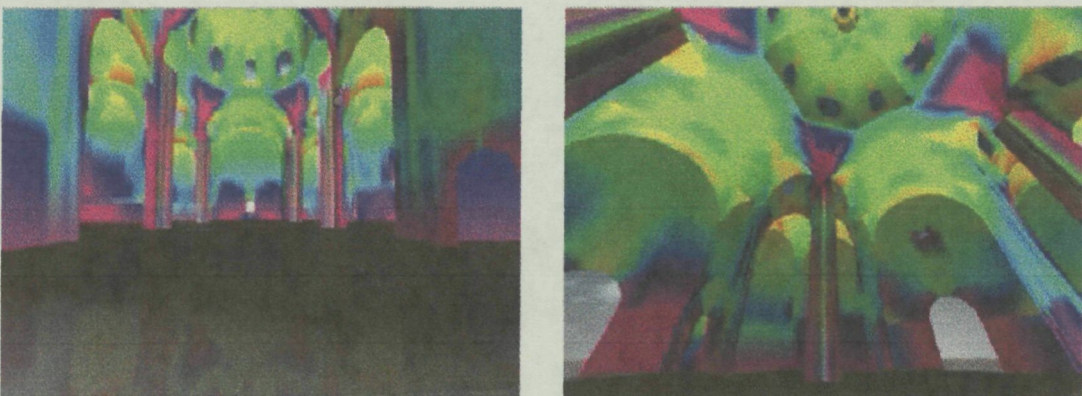


Fig. 5.1.14 Virtualización de cubiertas y elementos estructurales

Esta implementación a las interfaces desarrolladas en esta investigación, abre una nueva forma de análisis y conocimientos de la estructuras arquitectónicas, así como históricas. Es sabido que los edificios históricos tienen una geometría compleja y de difícil desarrollo. Para lo cual, esta investigación se presenta como una nueva herramienta para todos los profesionales que se dedican al estudio del comportamiento estructural y geométrico del acervo histórico-arquitectónico de nuestro mundo.

Figura 10.10. Estructura de la bóveda de la capilla de San Juan de los Rios, en un momento de la construcción. Fuente: [10].

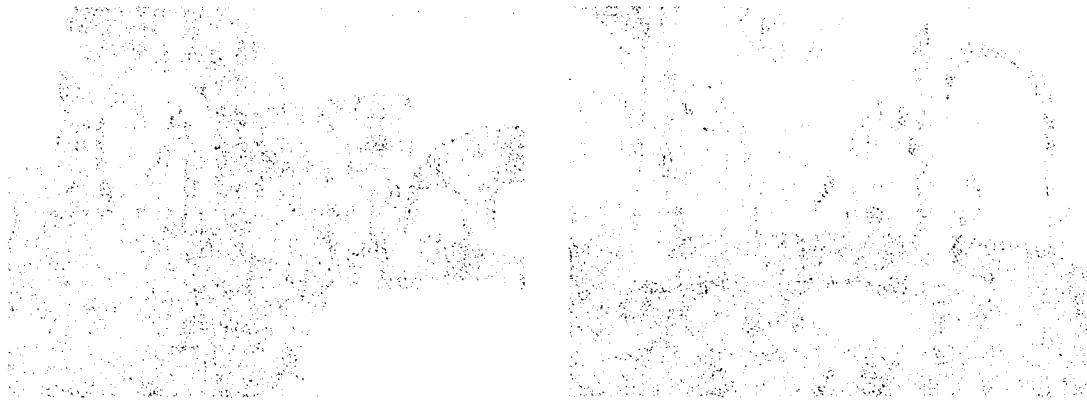


Figura 10.11. Estructura de la bóveda de la capilla de San Juan de los Rios, en un momento de la construcción. Fuente: [10].



Figura 10.12. Estructura de la bóveda de la capilla de San Juan de los Rios, en un momento de la construcción. Fuente: [10].

El modelo de la estructura de la bóveda de la capilla de San Juan de los Rios, en un momento de la construcción, se muestra en la Figura 10.13. El modelo de la estructura de la bóveda de la capilla de San Juan de los Rios, en un momento de la construcción, se muestra en la Figura 10.13. El modelo de la estructura de la bóveda de la capilla de San Juan de los Rios, en un momento de la construcción, se muestra en la Figura 10.13.

Figura 10.13. Estructura de la bóveda de la capilla de San Juan de los Rios, en un momento de la construcción. Fuente: [10].

El modelo de la estructura de la bóveda de la capilla de San Juan de los Rios, en un momento de la construcción, se muestra en la Figura 10.13. El modelo de la estructura de la bóveda de la capilla de San Juan de los Rios, en un momento de la construcción, se muestra en la Figura 10.13. El modelo de la estructura de la bóveda de la capilla de San Juan de los Rios, en un momento de la construcción, se muestra en la Figura 10.13.