ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Carrera

Desarrollo de "ANTWERP" (Application for Nets and Tessellations With Edge-to-edge Regular Polygons) e implementación visual online.

Development of "ANTWERP" (Application for Nets and Tessellations With Edge-to-edge Regular Polygons) and online visual implementation.

Para acceder al Titulo de

INGENIERO INDUSTRIAL

Autor: Sergio Díaz Gómez
Septiembre - 2015

TÍTULO	**DESARROLLO DE "ANTWERP" (APPLICATION FOR NETS AND TESSELLATIONS WITH EDGE-TO-EDGE REGULAR POLYGONS) E IMPLEMENTACIÓN VISUAL ONLINE**		
AUTOR	**SERGIO DÍAZ GÓMEZ**		
DIRECTOR / PONENTE	**VALENTÍN GÓMEZ JÁUREGUI**		
TITULACIÓN	***INGENIERÍA INDUSTRIAL***	**FECHA**	**SEPTIEMBRE 2015**

PALABRAS CLAVE: **MOSAICOS, MALLAS DOBLE CAPA, WEBGL, SOFTWARE, APLICACIÓN WEB.**

PLANTEAMIENTO DEL PROBLEMA

El propósito de este PFC es el de crear un Software capaz de generar automáticamente mosaicos y mallas de doble capa a partir de una nomenclatura propuesta por C.Otero (1994), demostrando que la misma es inequívoca y consistente, y ponerlo en uso obteniendo una malla de doble capa y analizándola desde un punto de vista estructural.

DESCRIPCIÓN DEL PROYECTO

"ANTWERP" es el acrónimo de "Application for Nets and Tessellations With Edge-to-edge Regular Polygons". Se trata de un software cuyo propósito original era producir automáticamente mosaicos según una nueva nomenclatura que fuese consistente y eficiente. Busca definir y generar mosaicos n-uniformes inequívocamente, compatibles con la nomenclatura actual usada para teselados regulares y arquimedeanos.

El código ha evolucionado a una segunda fase donde es posible generar mallas espaciales de doble capa (DLG, Double-Layer Grids). Para ello, la aplicación implementa las reglas de Otero modificadas, permitiendo el renombrado de las DLG existentes y la generación de nuevas mallas desconocidas hasta ahora.

Mediante algoritmos y códigos computacionales es posible recrear esas mallas 3D directamente, a partir de su nombre, y posteriormente visualizarlas.

Una vez la malla se muestra en pantalla, es posible realizar zoom y órbitas para cambiar perspectivas. También se añaden otras funcionalidades, como mostrar vistas principales, proyecciones, elementos auxiliares (ejes, numeración de nodos y barras…), y exportarlos para posteriormente ser utilizados en aplicaciones CAD/CAE.

El código fue originalmente diseñado y escrito mediante VBA, aplicado a AutoCAD. Sin embargo, se consideró más útil poder ejecutarlo en entornos web multi-plataforma, sin necesidad de tener ningún software instalado. Por ello, ANTWERP es ahora una aplicación web a la que se puede acceder desde cualquier entorno mediante una conexión de datos, mediante un PC o dispositivo móvil en un buscador que soporte WebGL (sobre JavaScript).

AGRADECIMIENTOS

Es la primera vez que me encuentro en la situación de tener que expresar por escrito mis agradecimientos; por ello, no seré breve.

En primer lugar, a mis padres y abuelos por todo su cariño y apoyo. Cada vez que escucho cómo hablan de mí con sus amigos, el orgullo que sienten…, siento que esa es y será siempre mi máxima motivación.

En segundo lugar, a mis amigos. Esos a los que les tuve que explicar varias veces “qué narices es ANTWERP?”. Porque me sirvió para tenerlo yo mismo mucho más claro.

A Valen por su paciencia y dedicación, su total disponibilidad e interés para con el proyecto. Él me lo propuso y cuando le dije que “quizás me venía un poco grande”, no lo tomó en serio.

A Aysu Şengünler, por cada palabra de aliento cuando faltaba el mismo.

Y también a Chester, por dejar de ladrar cuando se lo pedía.

A todos ellos, gracias. Muy sinceras.

RESUMEN

“ANTWERP” es el acrónimo de “Application for Nets and Tessellations With Edge-to-edge Regular Polygons”. Es un software cuyo propósito era producir automáticamente mosaicos según una nueva nomenclatura consistente y eficiente. Busca generar mosaicos n-uniformes inequívocamente, compatibles con la nomenclatura actual para mosaicos regulares y arquimedeanos.

El código ha evolucionado a una segunda fase donde es posible generar mallas espaciales de doble capa (DLG, Double-Layer Grids). Para ello, la aplicación implementa las reglas de Otero modificadas, permitiendo el renombrado de las DLG existentes y la generación de nuevas mallas desconocidas hasta ahora y por descubrir. Mediante algoritmos y códigos computacionales es posible recrear esas mallas 3D directamente, a partir de su nombre, y posteriormente visualizarlas.

Una vez la malla se muestra en pantalla, es posible realizar zoom y órbitas para cambiar perspectivas y otras funcionalidades, como mostrar vistas principales, proyecciones, elementos auxiliares (ejes, numeración de nodos y barras…), y exportarlos para posteriormente ser utilizados en aplicaciones CAD / CAE.

El código fue originalmente diseñado y escrito mediante VBA, aplicado a AutoCAD. Sin embargo, se consideró más útil poder ejecutarlo en entornos web multi-plataforma, sin necesidad de tener ningún software instalado. Por ello, ANTWERP es ahora una aplicación web a la que se puede acceder desde cualquier entorno mediante una conexión de datos, mediante un PC o dispositivo móvil en un buscador que soporte WebGL (sobre JavaScript).

El propósito de este PFC es el de generar dicho Software y ponerlo en uso obteniendo una malla de doble capa y analizándola desde un punto de vista estructural. Finalmente, se presenta como artículo al IASS 2015.

PALABRAS CLAVE: mosaico, malla doble capa, WebGL, aplicación web.

ABSTRACT

“ANTWERP” is the acronym of “Application for Nets and Tessellations With Edge-to- edge Regular Polygons”. Its main purpose was to prove that a new nomenclature for generating automatically tessellations was consistent and efficient. This proposal serves to define and generate n-uniform mosaics unequivocally, consistent with the current nomenclature used for Archimedean (regular and semirregular) tessellations.

The code has evolved up to a second phase where it is possible to generate double-layer grids (DLG). The application implements the modified Otero’s rules permitting the renaming of the existing DLG and the generation of new grids unknown until present. By means of an algorithm and some computational codes, it is possible to recreate in 3D any DLG directly from their own names. The program represents the design in the graphical window, interpreting the structure and generating the 3D geometry. It is possible to use the commands zoom or orbit to visualize it from different points of view. Other options are also available, like showing main views, projections, exporting data to txt, to be analyzed with other CAD/CAE tools.

The code was originally designed and written by means of user-defined functions (with Visual Basic for Application or VBA) applied to AutoCAD. However, the software could be more useful outside of the CAD environment as a web based application. Thus, ANTWERP is now a program that can be accessed over a network connection using HTTP, which means that the application can be run from any PC or handset device (tablet, phone...) on a web browser that supports WebGL (over JavaScript).

The purpose of this Final Degree Project is to create that Software and use it to obtain a DLG to analyze it from an structural point of view. Finally, it was presented as an article for the IASS 2015 Symposium.

KEY WORDS: tessellation, double layer grids, WebGL, web application.

ÍNDICE

1. INTRODUCCIÓN

Hoy en día, una de las principales dificultades en la configuración de mosaicos y mallas de doble capa (DLG; del inglés "Double Layer Grids") es su nomenclatura actual. Es un problema serio debido a la importancia de usar un nombre único para cada configuración geométrica diferente, tanto en 2D (mosaicos) como en 3D (mallas espaciales).

Las DLG son estructuras espaciales que contienen dos redes paralelas de miembros que forman la capa superior e inferior en forma de mosaico. Los nodos de cada capa están conectados mediante barras horizontales; las distintas capas quedan unidas por barras diagonales y/o verticales (Malla y Serrette 1996). Como ejemplo, entre ellos hay una tipología que es la más usada, común y extendida, compuesta por dos capas hechas a partir de cuadrados regulares. Sin embargo, aún siendo la red más común, no tiene una única denominación, sino varias y diferentes: "*Space-Deck", "Square-on-Square"*, etc. Para poder contemplar otras variaciones una serie de términos se añade a esta malla –u otros tipos de mallas-, tales como "offset", "set orthogonally", "set diagonally" y/o "hybrid chordal geometry". Sin embargo, esta clasificación no es suficientemente específica para incluir todas las variaciones de DLG existentes.

En relación a mosaicos, o teselados, la situación es diferente aunque también incongruente. A la hora de tratar con mosaicos compuestos por polígonos regulares, la notación de Cundy y Rollett (1981) es la forma más aceptada de describirlos y definirlos. Esta nomenclatura lista en orden horario y separado por puntos el número de lados de los polígonos que rodean cada vértice. Si hay varios números repetidos, un superíndice indica el número de repeticiones de dicho polígono (p.ej. $3.3.3.4.4=3^4 4^2$). Cundy y Rollett usaron esta notación sólo para mosaicos regulares y semirregulares (consistentes en polígonos regulares con diferente número de lados concurriendo en cualquier vértice y siempre respetando la misma configuración u otra análoga obtenida mediante reflexión,

rotación o traslación). Sin embargo, está aceptado mayoritariamente aplicarla también para mosaicos hemirregulares (donde la distribución de vértices no es única y sus polígonos regulares convergen en varias y diversas formas) listando cada tipo diferente de vértices separados por barras inclinadas (p.ej. 3.3.3.3.3.3/3.3.3.4.4/3.3.4.3.4=3^6 /3^3.4^2/3^2.4.3.4). Un ejemplo de este mosaico es el mostrado en la Figura 1.

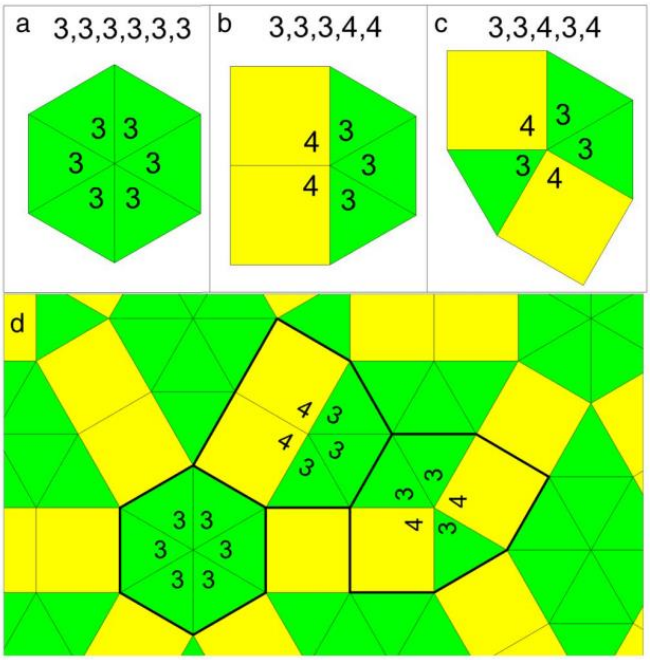


Figura 1. Composición de mosaico 3.3.3.3.3.3/3.3.3.4.4/3.3.4.3.4 (nomenclatura de Cundy y Rollet)

Esta notación tiene varios inconvenientes, como la necesidad de nombres demasiado extensos y poco intuitivos a la hora de definir los mosaicos. Aún más importante: estas denominaciones no son únicas y, de hecho las partículas #1, #2, etc., son necesarias para distinguir diferentes mosaicos con el mismo nombre. Esto se puede observar en el ejemplo del mosaico 3^6 /3^3.4^2/3^2.4.3.4, variantes #1 y #2 (números 11 y 12 de acuerdo a la numeración propuesta por Critchlow (1969)). Ambas están presentadas en la Figura 2.

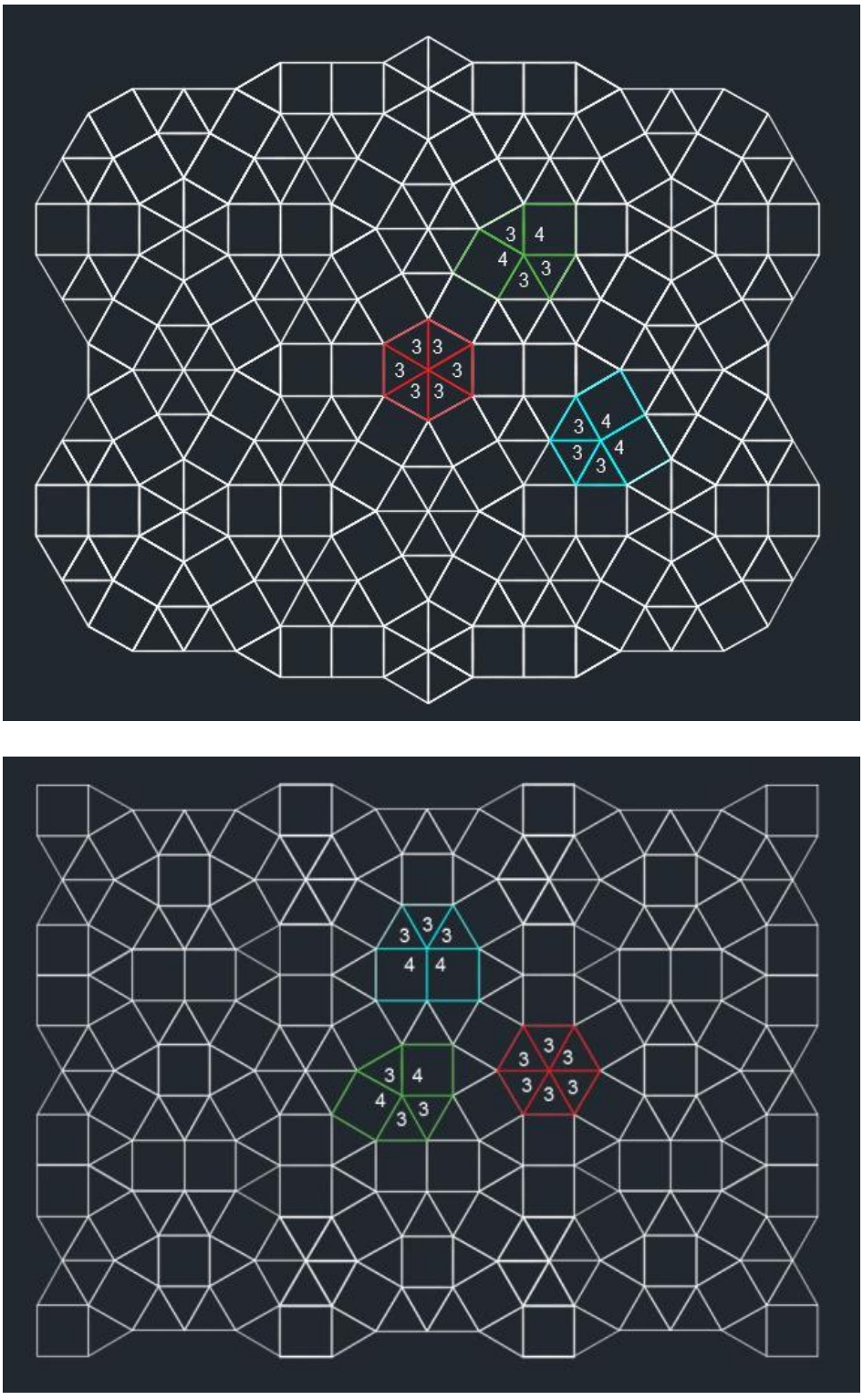


Figura 2. Mosaico 3^6 /$3^3.4^2$/$3^2.4.3.4$, variantes #1(arriba) y #2(abajo).

1.1- Objetivos y motivación

Una nueva nomenclatura para mosaicos y DLG se ha propuesto recientemente (Gómez-Jáuregui et. al. 1912). Establece una metodología sistemática para generar automáticamente diferentes mosaicos y DLG, con dos importantes ventajas:

1. Muestra cómo generarlas y diseñarlas a partir de los parámetros expresados en su propio nombre.
2. Define la notación de mosaicos y DLG de una forma sintetizada e inequívoca.

Por ejemplo, los nombres de los dos mosaicos mostrados en la Figura 2 serán totalmente distintos: 3-4-3,4-3,3,3/30/60 (para la variante #1) y 4-3,3-3,4-4,3,3-0,3,4,3/90/90 (para la variante #2).

Los mismos autores (2012) crearon una herramienta software: ANTWERP 1.0 (***Application for Nets and Tessellations with Edge-to-edge Regular Polygons***), capaz de representar en 2D cualquier tipo de mosaicos, programada en VBA para AutoCAD y disponible de forma gratuita.

El objetivo principal de este PFC ha sido desarrollar una segunda fase: llegar a generar dichos mosaicos y también DLG, aplicando las reglas de Otero (1990) modificadas , mediante un software fuera del entorno de AutoCAD, otorgándole mayor funcionalidad para usuarios no familiarizados con dicho programa; quedará probado por tanto que se trata de una nomenclatura eficiente y consistente. Se decidió crear para ello una aplicación Web (Diaz-Gomez et. al. 2015), accesible desde cualquier dispositivo con conexión a la red usando http: un PC, MAC, dispositivo móvil (Tablet, teléfono móvil…).

Por último, para demostrar el potencial de esta aplicación en un entorno ingenieril, se ha obtenido una malla espacial mediante este software y se han exportado los datos a ToyGL (software utilizado para el análisis de estructuras, explicado en profundidad más adelante), procesado posteriormente y mostrado

una malla con un buen equilibrio entre tracciones y compresiones.

1.2- Contenido de la memoria.

En esta memoria se expone, en una serie de capítulos, los pasos que han llevado a la consecución de los objetivos marcados.

- En el presente capítulo se hace una introducción, explican los contenidos y motivación, y se presenta el contenido de la memoria.

- En el capítulo 2 se analiza el estado del arte, la nueva nomenclatura propuesta para la generación de mallas, a partir de lo cual es posible determinar el proceso óptimo que nos lleve al desarrollo del software deseado, y la versión 1.0 de ANTWERP.

- En el capítulo 3 se describen todas las tecnologías y herramientas software que han sido utilizadas durante el desarrollo del proyecto.

- En el capítulo 4 se muestra una descripción general del software objeto de este proyecto, se realiza el análisis de requisitos de dicha aplicación software y se describen la metodología y diferentes partes del proceso de desarrollo .

- En el capítulo 5 se aborda la fase de pruebas del desarrollo del software y se explica en detalle cada una de las partes de la que consta.

- En el sexto capítulo se muestra su potencial en la generación de una malla determinada y su procesado mediante ToyGL.

- En el último capítulo, se incluyen las conclusiones obtenidas del trabajo realizado y se exponen diferentes posibles vías de ampliación o mejora del proyecto.

2. ESTADO DEL ARTE

En este capítulo se describirá la nueva propuesta de nomenclatura que se ha empleado para desarrollar la metodología de representación de las mallas y mosaicos.

Esta nomenclatura fue presentada originalmente en 2012. Sin embargo, desde entonces se han incluido algunas pequeñas modificaciones para poder llegar a representar un rango más amplio de composiciones.

Se describirá además la versión 1.0 de ANTWERP, usos y limitaciones.

2.1- Nomenclatura para mosaicos

Para explicar la metodología de forma sencilla, emplearemos un ejemplo de un mosaico 3-uniforme específico: la malla $3^6/3^3.4^2/3^2.4.3.4$ #1 según Cundy y Rollett (Figura 2, arriba). Según esta nueva nomenclatura, ese mosaico se llamará 3-4-3,4-3^2/30/60, donde cada guión (-) representa un cambio de fase, y las 2 barras inclinadas (/) representan los dos distintos tipos de simetría (siendo el primero la simetría del sector poligonal, y el segundo la simetría del módulo poligonal básico).

Por lo tanto, para este mosaico distinguimos 4 fases (Figura 3):

1.- La primera siempre corresponde al polígono semilla; en este caso es un triángulo (“3”). Figura 3.a.

2.- La segunda fase, en este caso, es un único cuadrado (“4”), que debe ser yuxtapuesto en la primera cara libre de la fase anterior, observándolas siempre en sentido horario. Figura 3.b.

3.- La tercera fase está compuesta por un triángulo y un cuadrado (“3,4”),

colocados en las caras libres de la fase 2, en el orden indicado según las agujas del reloj. Figura 3.c.

4.- En este caso, la cuarta fase será la última. Se corresponde con un dos triángulos (“3,3”), de nuevo colocados en las 2 primeras caras libres de la fase inmediatamente anterior, en sentido horario. Figura 3.d.

El conjunto formado por estas fases se denomina “sector poligonal”.

Después de la primera barra inclinada, “30” significa que el sector poligonal será reflejado por un eje que forma 30º con el eje de ordenadas. Figura 3.e. Esta simetría se repite continuamente, siendo para cada simetría el ángulo de referencia el doble del anterior; esto es, 60º la segunda simetría, 120º la tercera, y así sucesivamente hasta completar un mosaico de 360º. (Figura 3.f,g y h). Al terminar este proceso obtenemos lo que se conoce como “módulo poligonal básico”.

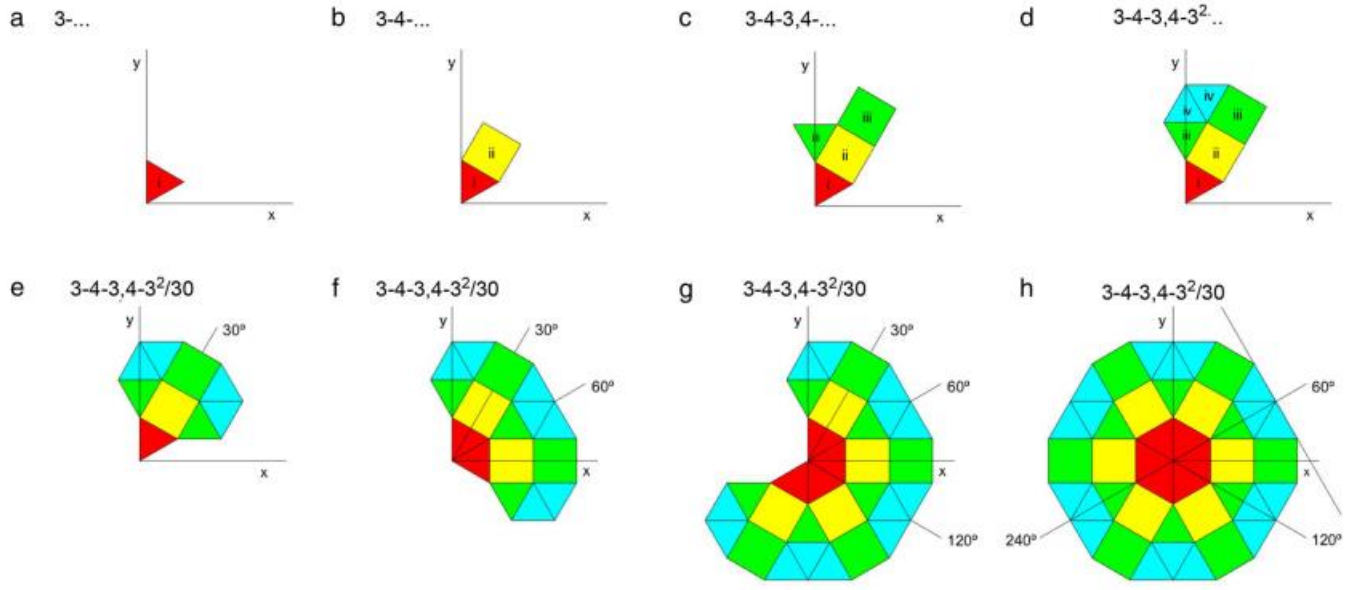


Figura 3: Generación de fases del módulo poligonal básico 3-4-3,4-3²/30/60

Finalmente, después de la segunda barra inclinada observamos “60”, que indica la inclinación de un eje a 60º con el eje de ordenadas. Este eje intersectará uno de los lados del módulo poligonal básico. Por esta intersección trazaremos un nuevo eje perpendicular al anterior, que servirá para localizar las simetrías del mosaico completo, sin crear nuevos polígonos solapados a la célula original. Repetiremos el mismo proceso con ángulos n veces el original (120, 180, 240…), siendo n un número natural ≥2, como se ve en la Figura 4.

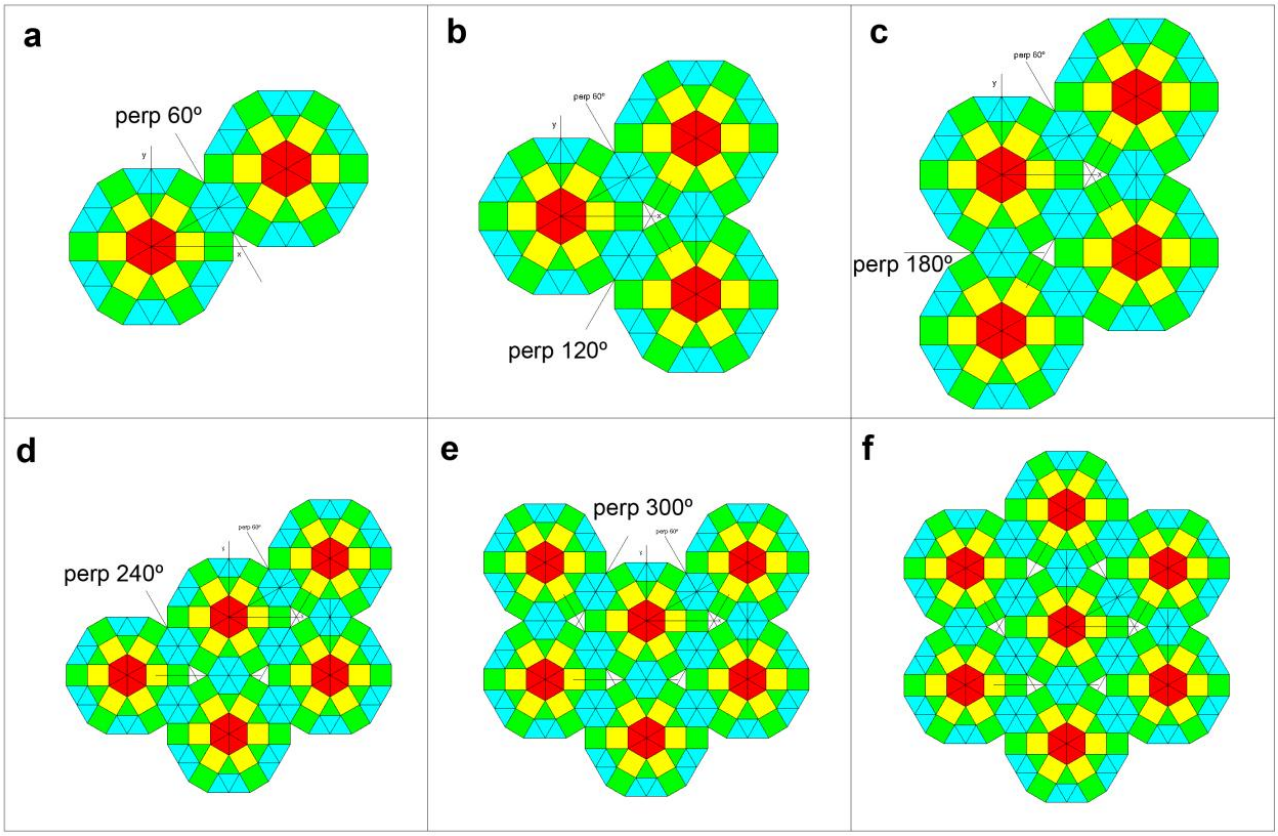


Figura 4: Generación de mosaico completo mediante las simetrías del módulo poligonal básico.

2.2- Nomenclatura para mallas de doble capa

En 1990, C. Otero propuso la generación de cualquier DLG empleando únicamente tres bloques de información:

1. El mosaico de barras diagonales.
2. Las reglas de alternancia que definen qué vértices están en cada capa.
3. La ley que determina la forma de unir los vértices que están incluidos en cada capa.

Este método fue implementado por Gómez-Jáuregui et al. (2012) añadiendo nuevos parámetros y condiciones para aumentar el número de soluciones e incrementar la flexibilidad de esta técnica particular.

Una de las mejoras es la posibilidad de aplicar las reglas 2 y 3 en cada una de las capas, independientemente. Haciendo esto, sería posible formular las leyes en cada capa de la siguiente forma:

e) “everyone”: cada vértice del mosaico pertenece a esa capa en particular.

a) ”alternated”: los vértices pertenecerán o no a esa capa, alternativamente.

m) “middle”: los puntos medios entre vértices del mosaico pertenecen a esa capa en particular.

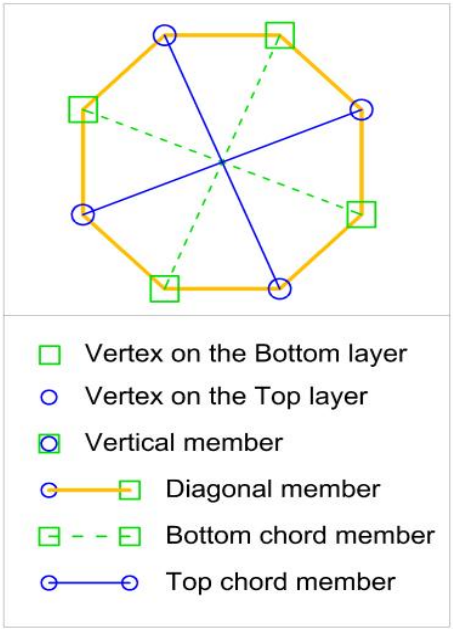


Figura 5. Set de símbolos para las DLG

En relación con la tercera ley, determinando la forma de conectar los vértices en cada capa, se propuso añadir una definición simbólica al símbolo de la segunda ley. (Ver Figura 6)

1) Unir cada vértice con el más cercano de cada polígono.
2) Unir cada vértice con el segundo más cercano de cada polígono.
3) n –Unir cada vértice con el n más cercano de cada polígono.

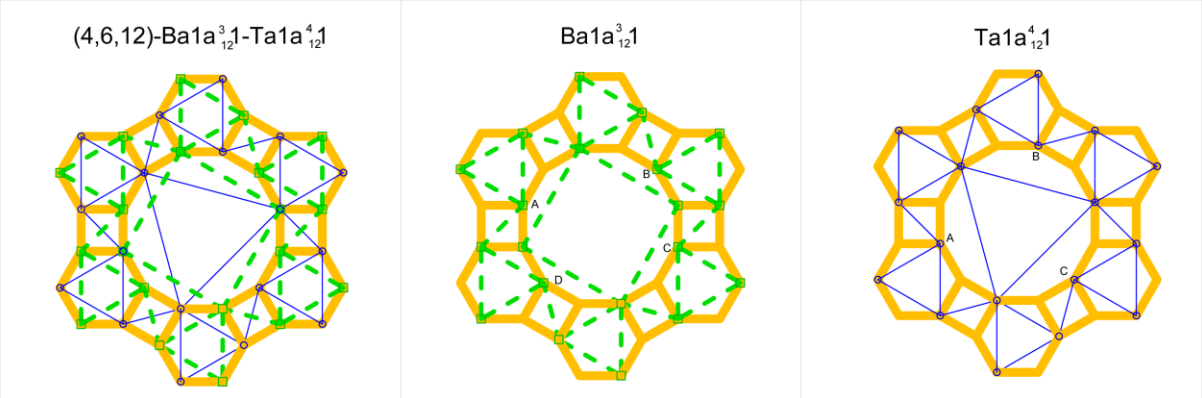


Figura 6: Ley alterna para colocar los vértices con frecuencias tres (a3) y cuatro (a4).

2.3- ANTWERP v1.0

El código fue diseñado originalmente mediante funciones definidas por el usuario (con Visual Basic Application, "VBA") e implementadas en AutoCAD. Permitía, ejecutando una simple "macro", lanzar una ventana de diálogo que servía como interfaz básica (Figura 7). Existía la posibilidad de generar mosaicos totalmente nuevos u otros ya existentes, presentes en 3 menús de selección desplegables, incluyendo Mosaicos Uniformes (o Arquimedeanos), 2-uniformes y k-uniformes. Además, existían otras opciones relacionadas con la visualización del módulo poligonal básico (mini-mosaico), nombrado del mosaico, numeración de lados, ejes de referencia coloreados, líneas auxiliares, etcétera.

El hecho de que fuera originado para ser usado en AutoCAD podría limitar su funcionalidad a usuarios familiarizados con dicho software, y además, hacer que sólo fuera posible su visualización en ordenadores, que además deberían tener instalado AutoCAD.

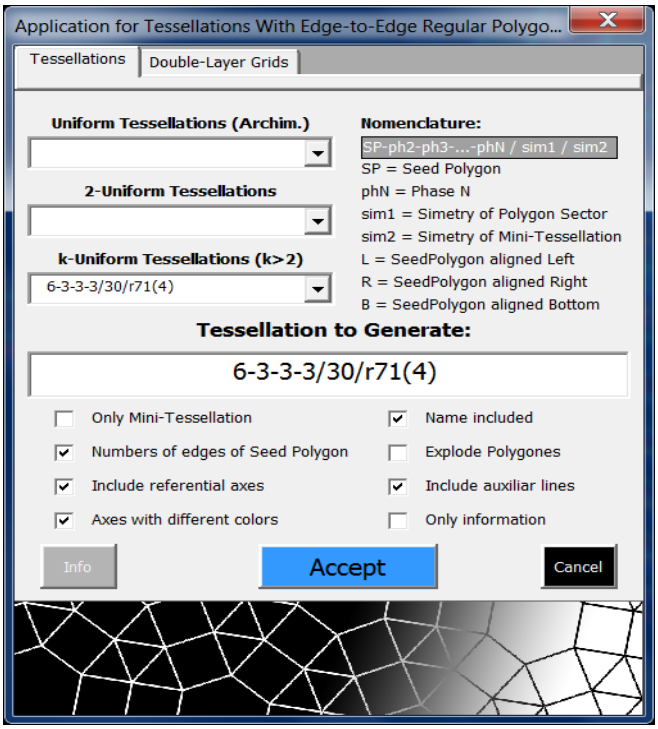


Figura 7: Menú de ANTWERP v1.0

3. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS.

En este tercer capítulo se explica cuáles han sido las herramientas, tecnologías y lenguajes de programación empleados a lo largo del desarrollo del proyecto, identificando su aportación para con el mismo y el motivo de su elección.

3.1- Lenguajes y tecnologías empleadas

3.1.1- HTML

De acuerdo a la W3C (Organización dedicada a la estandarización de tecnologías ligadas a la Web, su estructura, escritura e interpretación), podemos definir HTML -siglas de ***HyperText Markup Language*** («lenguaje de marcas de hipertexto»)- como el lenguaje para describir la estructura de páginas web. HTML proporciona a los autores medios para publicar documentos online con textos, tablas, listados, fotos, videos…, recopilar información mediante hipertextos, etc. Es un estándar de referencia para la elaborar páginas web en sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web. Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, *script*, ...), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene sólo texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que

cualquier página web escrita en una determinada versión pueda ser interpretada de la misma forma por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.). No obstante, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados.

Es muy fácil de utilizar y empleado universalmente, lo que ha llevado a la decisión de emplear este soporte.

3.1.2- JavaScript

JavaScript (Flanagan 2010) (también conocido como “JS”) es un lenguaje de programación interpretado, que se define como orientado a objetos, imperativo, dinámico, basado en prototipos y débilmente tipado. Generalmente es usado para definir la ejecución de tareas en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo una mejor gestión de la interfaz de usuario de páginas web dinámicas.

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, siendo éste uno de los motivos de peso para elegir el mismo en este proyecto.

Actualmente JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Figura 8: Logotipo de JavaScript

3.1.2.1- WebGL

WebGL es una especificación estándar que está siendo desarrollada actualmente para mostrar gráficos 3D en navegadores web. El WebGL permite mostrar gráficos 3D acelerados por hardware (GPU) en páginas web, sin la necesidad de plug-ins en cualquier plataforma que soporte OpenGL 2.0. Se utiliza el elemento canvas HTML5 y se accede mediante interfaces Document Object Model.

WebGL carece de las rutinas matemáticas matriz eliminadas en OpenGL 3.0. Esta funcionalidad debe ser proporcionada por el usuario en el espacio de código JavaScript.

Como WebGL es una tecnología diseñada para trabajar directamente con el GPU (unidad de procesamiento gráfico) es difícil de codificar en comparación con otros estándares web más accesibles, es por eso que muchas bibliotecas

de JavaScript han surgido para resolver este problema: SceneJS, SpiderGL, TDL, Three.js, X3DOM. BabylonJS…

Entre ellas Three.js es la más popular en términos de número de usuarios. Es ligera y tiene un bajo nivel de complejidad en comparación con la especificación WebGL original; es la que se ha empleado en el proyecto por este motivo

Figura 9: Logotipo de WebGL

3.1.2.2- ThreeJS

Es una biblioteca liviana escrita en JavaScript para crear y mostrar gráficos animados por ordenador en 3D en un navegador Web y puede ser utilizada en conjunción con el elemento canvas de HTML5, SVG ó WebGL. El código fuente está alojado en un repositorio en GitHub.

Se ha popularizado como una de las más importantes para la creación de las animaciones en WebGL.

Bibliotecas de alto nivel tales como Three.js o GlgE, SceneJS, PhiloGL u otras, permiten al autor complejas animaciones 3D que se muestran en el navegador sin el esfuerzo que se requiere para una aplicación independiente tradicional con un plugin.

3.2- Herramientas

3.2.1- Microsoft Visual Studio

Microsoft Visual Studio es un entorno (IDE, por sus siglas en inglés “Integrated Development Environment”) para sistemas operativos Windows; un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y hace más sencilla la creación de soluciones en varios lenguajes Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic, F#, Java, JavaScript, Python, Ruby, PHP…

Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML.

3.2.2- Trello

Trello es una aplicación de gestión de proyectos basada en la web, creada por Fog Creek Software (2011). Opera según un modelo de negocio “freemium”; un servicio básico y gratuito con una opción “Premium” de pago. Se ha utilizado en este proyecto para coordinar las fechas entre alumno-profesor, llevar un orden de prioridades y de avances.

3.2.3- ToyGL

Desarrollado por Julien Averseng et al. (2012), ToyGL es un software gratuito de modelado y cálculo de esfuerzos en estructuras espaciales, empleado para analizar finalmente en este PFC una malla obtenida con ANTWERP.

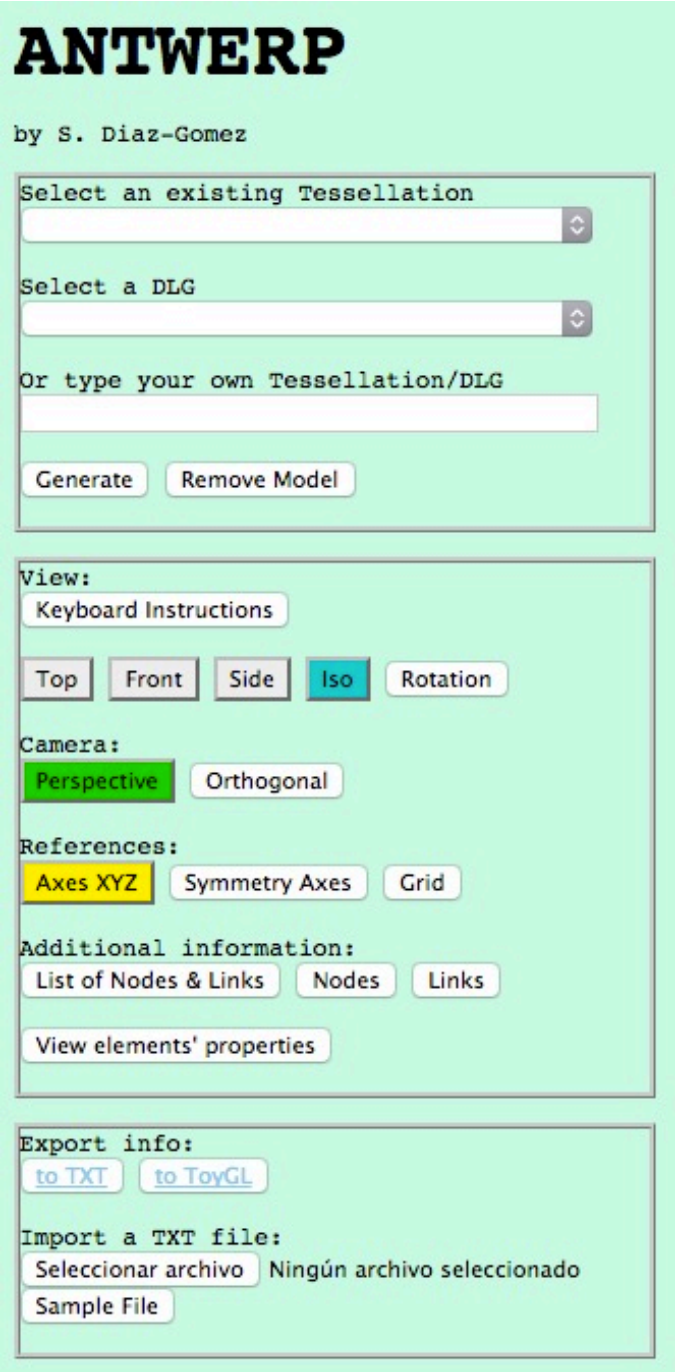
4. DESCRIPCIÓN DEL TRABAJO Y METODOLOGÍA

En este capítulo se realiza una descripción general de las funcionalidades y el comportamiento del software. Se parte de un análisis de los requerimientos que debe cumplir el mismo para lograr satisfacer al usuario final y generar una aplicación robusta. A continuación se plantea la arquitectura funcional con la que dar respuesta a estas necesidades y se desarrolla la metodología aplicada para hacerla realidad.

4.1- Descripción general de la aplicación

El propósito de ANTWERP es el de generar automáticamente cualquier mosaico y malla de doble capa, a partir de un nombre normalizado, inequívoco e intuitivo. La versión 2.0, elaborada en este proyecto, se basa en parte de los algoritmos de la versión 1.0, principalmente para crear las fases originales y aplicar los diferentes tipos de simetrías. Sin embargo las diferencias de programación entre VBA y JavaScript son notables. Esto ha llevado a la modificación de gran parte de los algoritmos y la inclusión de nuevas rutinas, principalmente para la generación de mallas de doble capa.

El hecho de que fuese originalmente creado para correr sobre AutoCAD podría limitar la funcionalidad de ANTWERP a usuarios familiarizados (y poseedores) de dicho programa. Sin ninguna duda, había que liberar al software de estas restricciones: sacarlo del entorno CAD y transformarlo en una aplicación web. Por ello, ANTWERP es ahora un software al que se puede acceder desde una red utilizando una conexión http, lo que significa que puede ser empleado desde cualquier PC o dispositivo móvil en cualquier navegador web que soporte WebGL.



Los menús se han escrito en inglés para facilitar su uso internacionalmente.

La interfaz gráfica del usuario está diseñada buscando una gran sencillez.

Figura 10: Menús en ANTWERP 2.0

Es posible generar diferentes tipos de mosaicos conocidos (regulares, semirregulares y hemirregulares o n-uniformes), seleccionándolos desde un menú desplegable mostrado en la imagen, o introducir manualmente el nombre de un mosaico a generar. Una de las mayores diferencias con la versión 1.0 es la posibilidad de hacer lo mismo con DLG.

Existe la posibilidad de exportar la posición de nodos y barras calculados en la aplicación a un formato TXT o adaptado a ToyGL para un procesado posterior.

En cuanto a las opciones de visualización, contamos con diferentes botones para cambiar las vistas (alzado, planta y perfil), el sistema de representación (isométrico y ortogonal), y generar una rotación del sistema, tal y como se observa en la Figura 10.

También es factible mostrar y ocultar los ejes del sistema de representación, y los ejes de simetría utilizados en cada generación, además un mallado en la base.

Otra funcionalidad muy interesante es la de mostrar u ocultar el número de cada nodo y barra, tanto en pantalla, como en una lista con sus posiciones y otros atributos.

Se consideró la opción de ir mostrando paso a paso la generación de los mosaicos en pantalla (Figura 11); esto sería de gran valor académico, pero probablemente su valor industrial no se vería afectado directamente y la velocidad de ejecución quedaría enormemente disminuida. En la versión final esta idea quedó desechada.

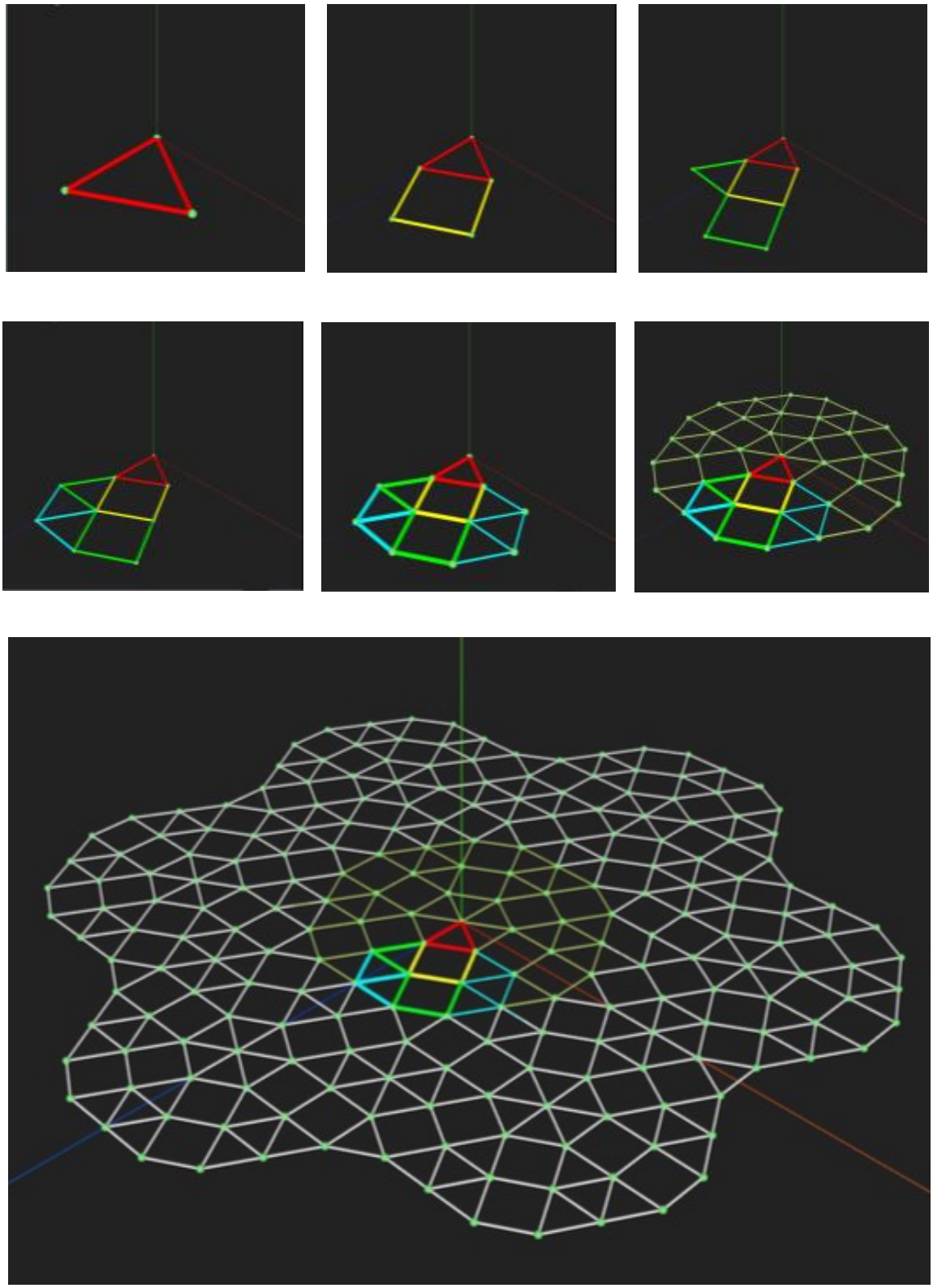


Figura 11: Generación paso a paso de la malla 3-4-3,4-3^2/30/60 en ANTWERP v2.0

4.2- Análisis de requisitos

Tras realizar la descripción general del funcionamiento del software, a continuación se incluyen los requisitos identificados a partir del uso planteado y que deben cumplirse al final del desarrollo.

4.2.1- Requisitos funcionales

Id	Descripción del Requisito
RF01	El usuario podrá seleccionar mosaicos a partir de un menú desplegable.
RF02	El usuario podrá seleccionar DLG a partir de un menú desplegable.
RF03	El usuario podrá introducir un nuevo elemento a representar, según su nombre, en un cuadro de diálogo.
RF04	El usuario podrá limpiar la ventana gráfica en cualquier momento.
RF05	El usuario podrá seleccionar diferentes vistas y realizar órbitas y zooms.
RF06	El usuario podrá ver los ejes del sistema de representación, y de simetría, independientemente.
RF07	El usuario podrá ver la numeración de nodos y barras independientemente.
RF08	El usuario podrá exportar la lista de elementos calculados en formato .txt o adaptado a ToyGL

Figura 12: Tabla de requisitos funcionales de la aplicación.

4.2.2- Requisitos no funcionales

Id	Descripción del Requisito	Categoría
RNF01	El software deberá ser capaz de representar cualquier elemento, siempre que esté diseñado de forma coherente	Seguridad, Eficiencia.
RNF02	El software deberá ser ejecutable en cualquier dispositivo con WebGL independientemente de su resolución.	Accesibilidad, Portabilidad.
RNF03	El software deberá exportar los datos en un formato estándar y adaptado a ToyGL.	Portabilidad.
RNF04	Las barras deberán ser distinguibles fácilmente dependiendo de su familia.	Eficiencia
RNF05	El software deberá mostrar en todo momento el nombre del elemento que está siendo mostrado en pantalla.	Eficiencia.
RNF06	El software deberá asegurar que los “atajos de teclado” sólo sean efectivos al estar el mouse sobre la ventana de WebGL.	Accesibilidad.

Figura 13: Tabla de requisitos no funcionales de la aplicación.

4.3- Metodología

En todo proyecto software suele existir una metodología de trabajo inherente al mismo en la que se definen las tareas a realizar y las etapas de las que consta el proceso de desarrollo y en las cuales se llevan a cabo dichas tareas. El objetivo de fijar una metodología de este tipo es mejorar la productividad durante el desarrollo del software y, por consiguiente, la calidad final del producto.

En este proyecto concreto la metodología utilizada ha sido la incremental o iterativa. Se empieza con una versión muy primaria de la aplicación (en la primera iteración), a la que se le van añadiendo funcionalidades nuevas y/o mejorando apartados ya existentes en cada una de las iteraciones posteriores, consiguiendo así diferentes versiones del software final con cada nueva iteración.

Tras cada versión, ésta es evaluada, probada y estudiada, y se anotan posibles mejoras o actualizaciones que el software pudiera implementar en sus siguientes versiones, de forma que el producto final cumpla tanto con todos los objetivos y requerimientos que se plantearon al comienzo del desarrollo, como con aquellos que han sido generándose como consecuencia de la evaluación de alguna versión intermedia.

En la Figura 14 se puede observar un esquema gráfico de la metodología utilizada.

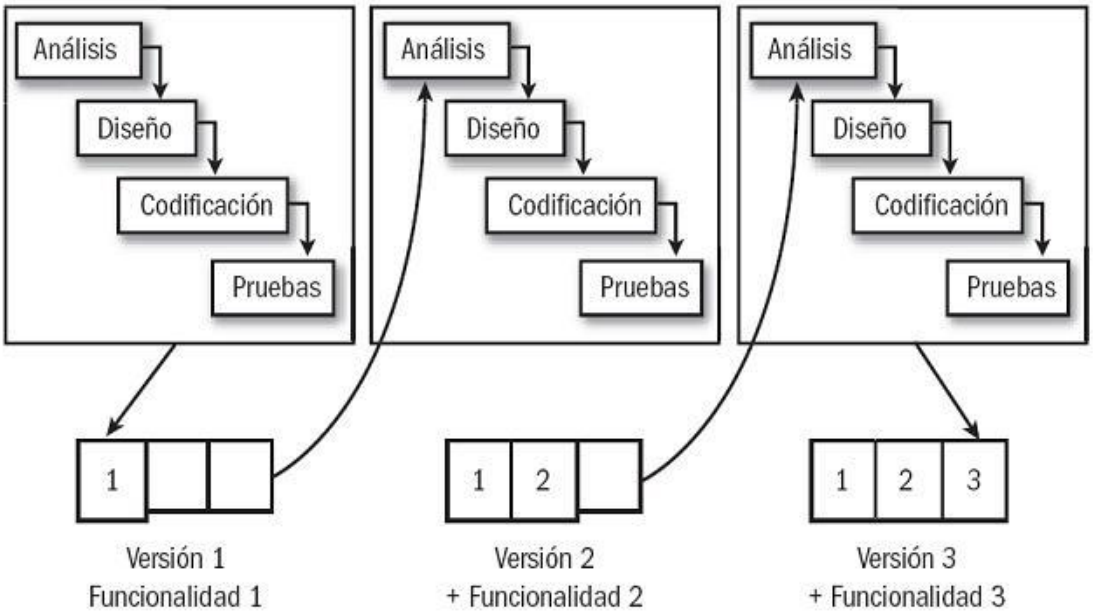


Figura 14: Proceso de desarrollo del software.

Utilizando la metodología descrita, durante este proceso de desarrollo se han realizado las siguientes iteraciones:

. Diseño primario de la interfaz y de los menús.

. Creación de los algoritmos de cálculo geométrico de mosaicos.

. Creación de los algoritmos de cálculo específicos para DLG.

. Creación de las funciones de exportación y opciones de visualización.

. Retoques finales de interfaz y gráficos.

Como se ha mencionado anteriormente, en cada una de las iteraciones también se repasaba cada una de las anteriores y se iban haciendo cambios y modificaciones desde la primera hasta la última, que afectaban a las superiores.

4.4- Diseño y programación del software

4.4.1- Menú

El menú de ANTWERP ha sido diseñado buscando una máxima sencillez y limpieza, mostrando sólo las opciones fundamentales. Ocupando un ancho de un 20% del total de la página y el 100% del alto, se encuentra dividido en tres bloques principales, cada uno correspondiente a un tipo de funcionalidad.

En el primer bloque (Figura 15), cuya misión es la de determinar la malla o mosaico a mostrar, encontramos dos botones desplegables –el primero para mosaicos y el segundo para mosaicos-, además de un cuadro de texto en el que el usuario puede introducir directamente el nombre de mosaico o malla que

desee (ANTWERP determinará automáticamente si se trata de un mosaico o de una malla y los representará).

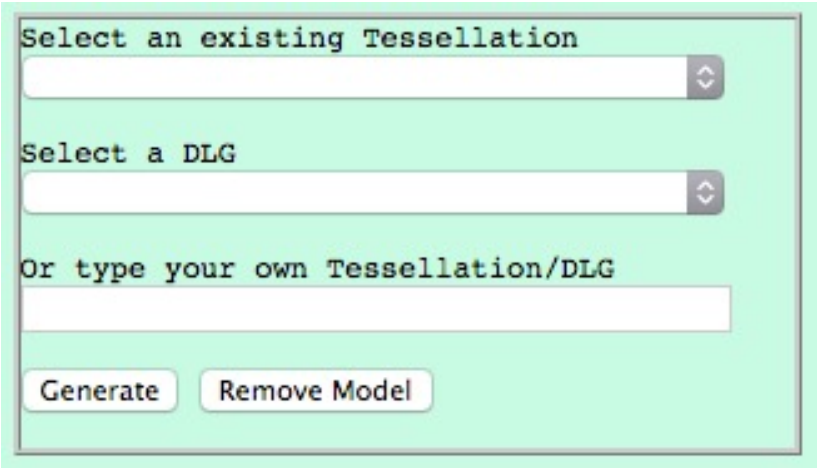


Figura 15: Menú de selección.

En el segundo bloque (Figura 16) se encuentran las funciones de visualización. Podremos desplegar las instrucciones para usar el teclado, seleccionar la vista deseada, si se quiere que la malla rote espacialmente, el tipo de perspectiva –ortogonal o isométrica-, y las referencias (permitiendo mostrar u ocultar ejes del sistema, ejes de simetría y el mallado de referencia en el plano de tierra).

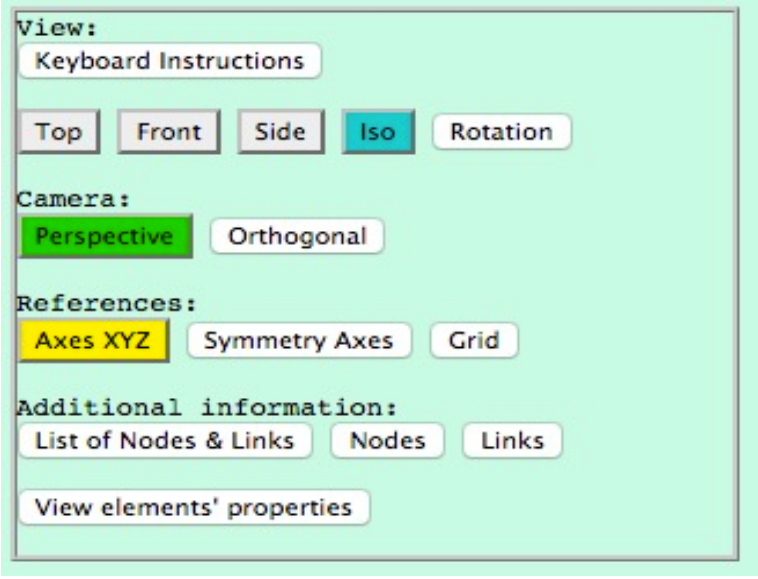


Figura 16: Menú de visualización.

En caso de solicitarlo, también se puede indicar la numeración de los nodos y de las barras, independientemente, y mostrar el listado de los mismos en pantalla.

Por último, dentro de este bloque de menú podremos encontrar una función para mostrar el análisis morfológico de la malla/DLG representada. Figura 17.

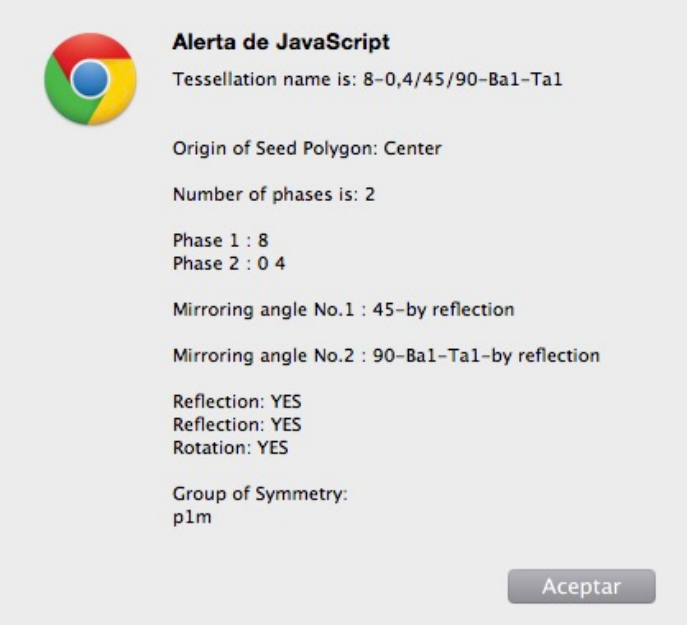


Figura 17: Análisis morfológico del mosaico

En el tercer y último bloque del menú (Figura 18) encontramos las opciones para exportar e importar datos. La primera nos permite exportar el listado de nodos –número total, índice y coordenadas de cada uno-, y barras –origen, final, familia-, en TXT para un fácil análisis, o pre-formateado para ser importado posteriormente en ToyGL.

La opción de importar nos permite utilizar ANTWERP como un visualizador en el que representar cualquier elemento tridimensional formado por nodos y

barras, siempre y cuando estén en un pre-formateado que ANTWERP pueda interpretar.

Figura 18: Menú de Datos.

4.4.2- Ventana de representación

La ventana de representación (Figuras 19 y 20), generada mediante un módulo de WebGL, ocupa un 80% del ancho de la ventana del navegador, y un 100% del alto. Está diseñada para mostrar mediante diferentes colores las barras que componen cada fase, en caso de estar representando mosaicos; si representamos mallas de doble capa, habrá un color para las barras en cota cero, otro color para las barras diagonales, y otro color para las barras de la capa superior.

En el centro superior, se representa en todo momento el nombre del elemento generado

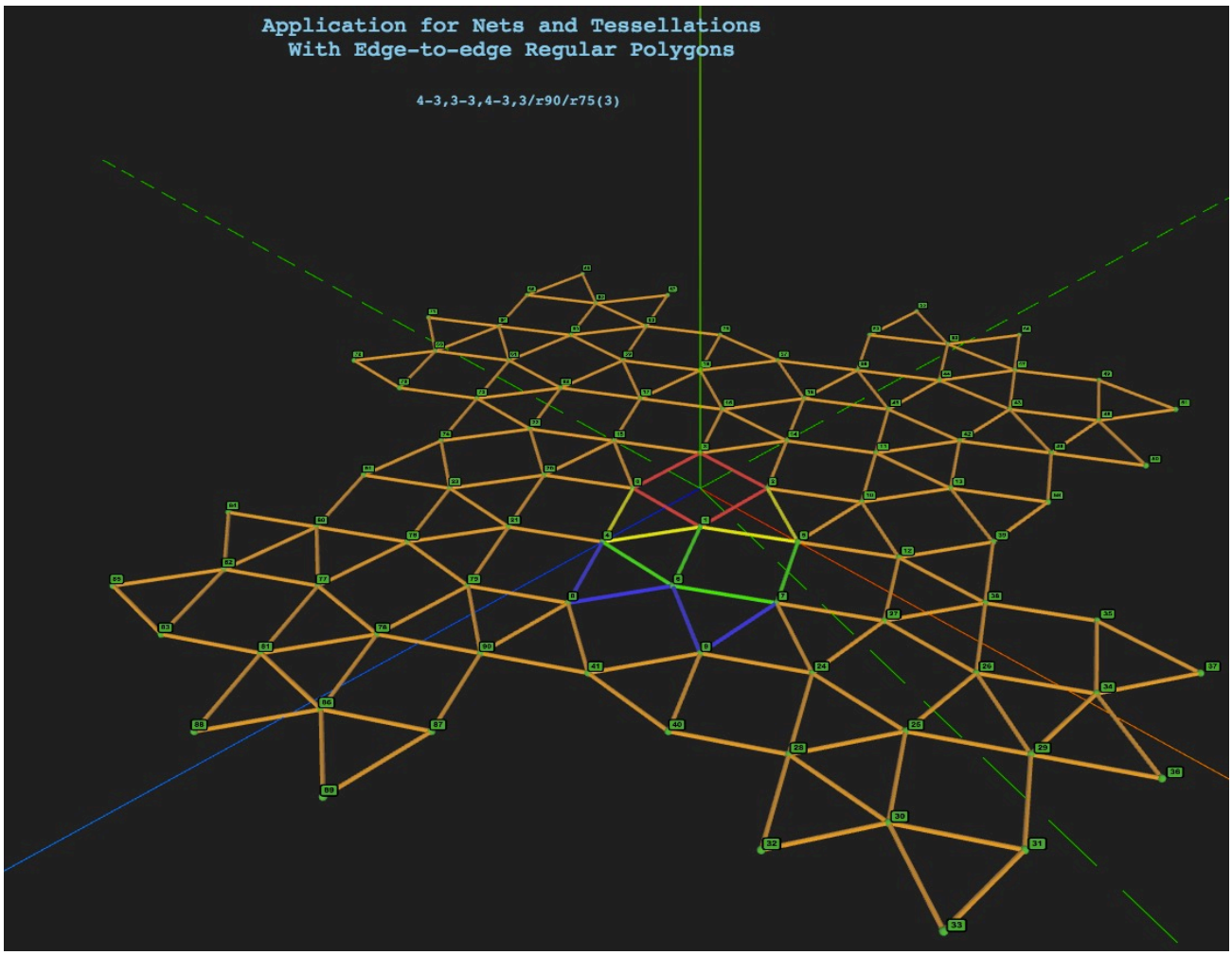


Figura 19: Representación de malla 4-3,3-3,4-3,3/r90/r75(3) e indicación por colores de fases, numeración de nodos, ejes de referencia y auxiliares para simetría.

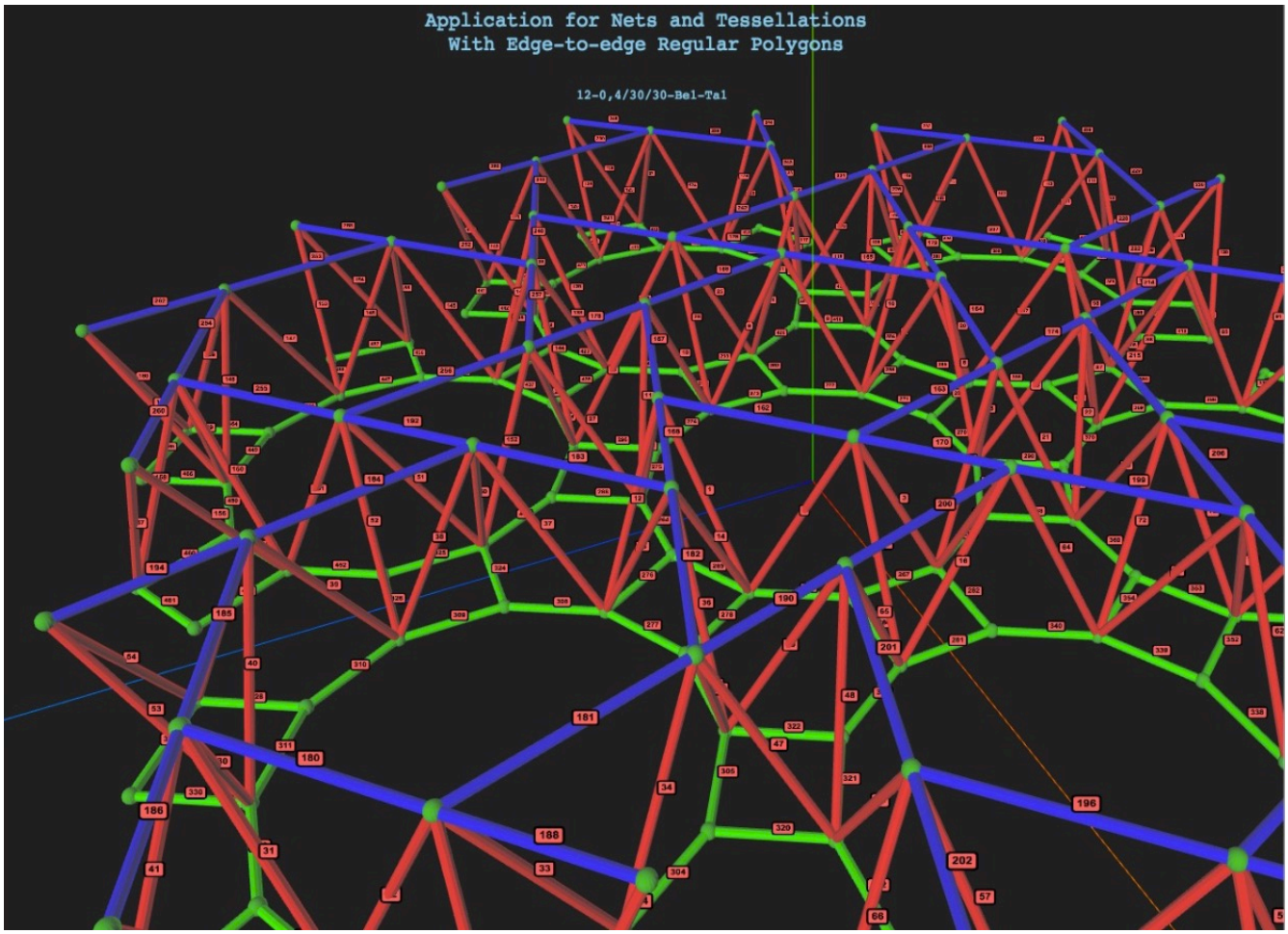


Figura 20: Representación de DLG12-0,4/30/30-Ba1-Ta1 e indicación por colores de la las capas de barras, numeración de barras y ejes de referencia.

5. PRUEBAS

Durante el proceso de desarrollo de cualquier herramienta software se hace necesario añadir una fase de validación y pruebas que permita garantizar el correcto funcionamiento de la aplicación, así como su seguridad e integridad.

Como se ha mencionado en el capítulo 4, durante todo el proceso de desarrollo se ha utilizado la metodología incremental o iterativa, que obliga a realizar, tras cada iteración del trabajo, una serie de verificaciones y pruebas que sirvan para garantizar el correcto desarrollo de la aplicación.

El objetivo fundamental de estas pruebas es comprobar que todo lo implementado funciona correctamente en cada iteración. Así, se hace necesario probar tanto las nuevas actualizaciones realizadas en la última iteración, como todas las anteriores, y la interacción directa entre todas ellas.

De esta forma, y como es costumbre a la hora de realizar este tipo de pruebas, éstas se han realizado dividiéndose en cuatro tipos:

. Pruebas unitarias.

. Pruebas de integración.

. Pruebas de sistema.

. Pruebas de aceptación (omitidas al requerir público familiarizado en la materia)

5.1- Pruebas unitarias

Las pruebas unitarias se basan en la verificación independiente de cada módulo del sistema. En cada iteración se han ido probado las nuevas actualizaciones concretas que se han ido añadiendo, ya sean parte de la

interfaz, del cálculo de nodos, de su conexión con barras, o de la exportación de datos.

En estas pruebas se ha hecho especial hincapié en los puntos críticos de cada apartado, realizando pruebas concretas buscando el error. Así, se han hecho verificaciones de botones concretos, de diferentes tipos de visualización, de los formatos de exportación, etc.

5.2- Pruebas de integración

Con las pruebas de integración se pretende, una vez verificados los módulos unitarios por separado, ampliar el rango de alcance de las pruebas de forma que se compruebe el comportamiento de dichos módulos a la vez que se van integrando en el sistema con cada iteración.

Así, se ha ido probando que tras cada iteración, todo lo introducido en el sistema funcionaba correctamente. Estas pruebas han consistido en probar la coherencia entre menús de selección y elementos representados, el correcto funcionamiento de los botones de visualización –y atajos del teclado- sobre el módulo de WebGL, la coherencia entre lo representado y las listas de datos exportadas ,etc.

5.3- Pruebas del sistema

Estas pruebas, finalmente, son las que se encargan de verificar que todo el sistema funciona correctamente en su conjunto, con todos los módulos y funciones instaladas, como parte final de todas las iteraciones realizadas.

Durante esta última fase de pruebas se comprueba el cumplimiento de todos los requisitos planteados, tanto funcionales como no funcionales. Así, se verifican también aquellos requisitos más relacionados con aspectos de

seguridad, rendimiento o accesibilidad.

De esta forma se han hecho pruebas portando el software a diferentes dispositivos como PC con diferentes sistemas operativos y exploradores, smartphones y tablets, con diferentes tamaños y hardware. De esta forma se ha podido comprobar que el software funciona de manera correcta en todos ellos.

6. OBTENCIÓN Y ANÁLISIS DE UNA DLG.

Para demostrar la aplicación práctica de ANTWERP, en esta parte del PFC se ha decidido realizar el proceso para determinar una DLG (malla de doble capa) de utilidad ingenieril, obtener la posición de sus nodos y barras en el espacio, exportar la información de los mismos en un pre-formato apropiado para ToyGL y analizarla con dicho software.

La malla seleccionada, por su atractiva geometría en una aplicación ingenieril -por ejemplo, como una cubierta- es la 4-8/45/90-Ba1-Ta1.

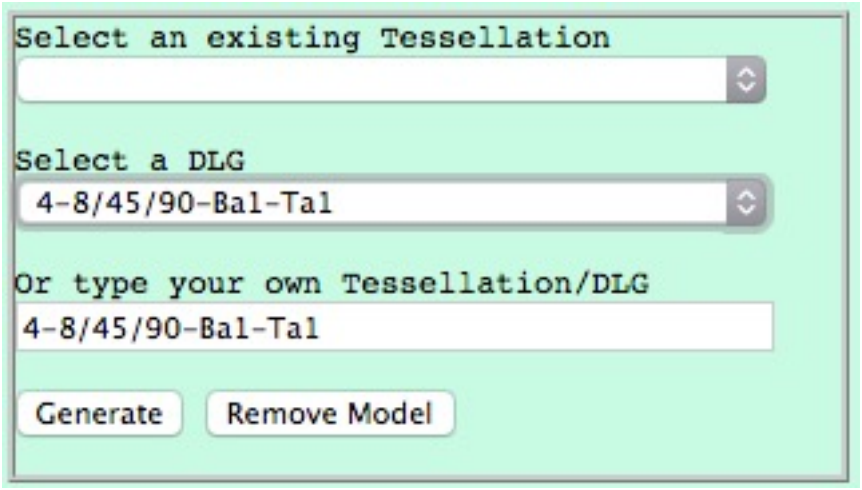


Figura 21: Selección de DLG 4-8/45/90-Ba1-Ta1 en ANTWERP.

Una vez seleccionada, ANTWERP procesará el nombre y la generará en la ventana gráfica de WebGL, tal como indica la Figura 22.

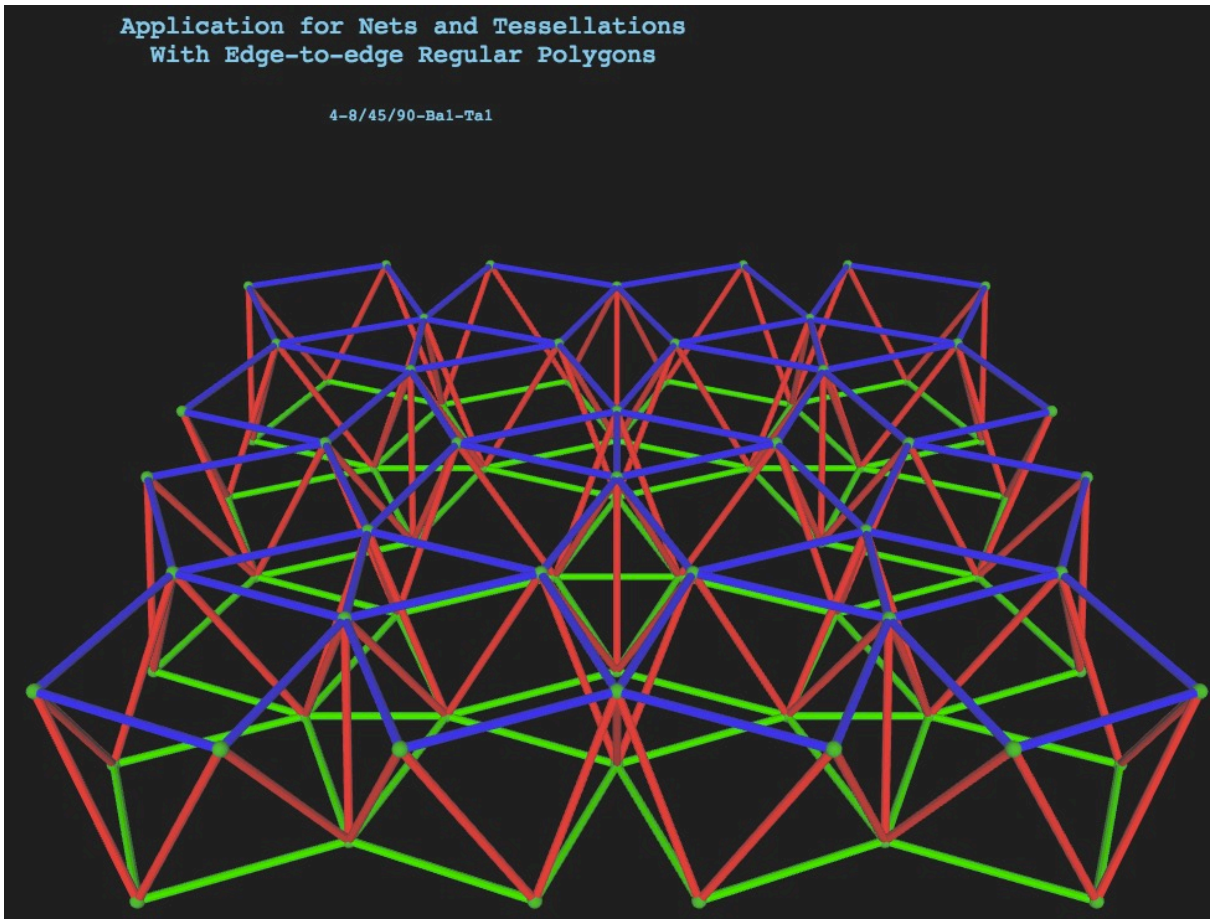


Figura 22: Representación Isométrica de DLG 4-8/45/90-Ba1-Ta1 en ANTWERP.

Al observar detenidamente la vista en planta (perspectiva ortogonal, Figura 23 superior) de esta DLG, se percibirá la similitud con el mosaico 4-8/45/90, a partir del cual se determina la posición de las barras de la capa diagonal, que dará lugar a la malla de doble capa elegida (Figura 22). La capa superior e inferior vendrá determinada por la información contenida en “Ba1-Ta1”; cada nodo pertenecerá alternativamente a la capa inferior o a la superior, y en cada una, se unirán al nodo más cercano de uno en uno

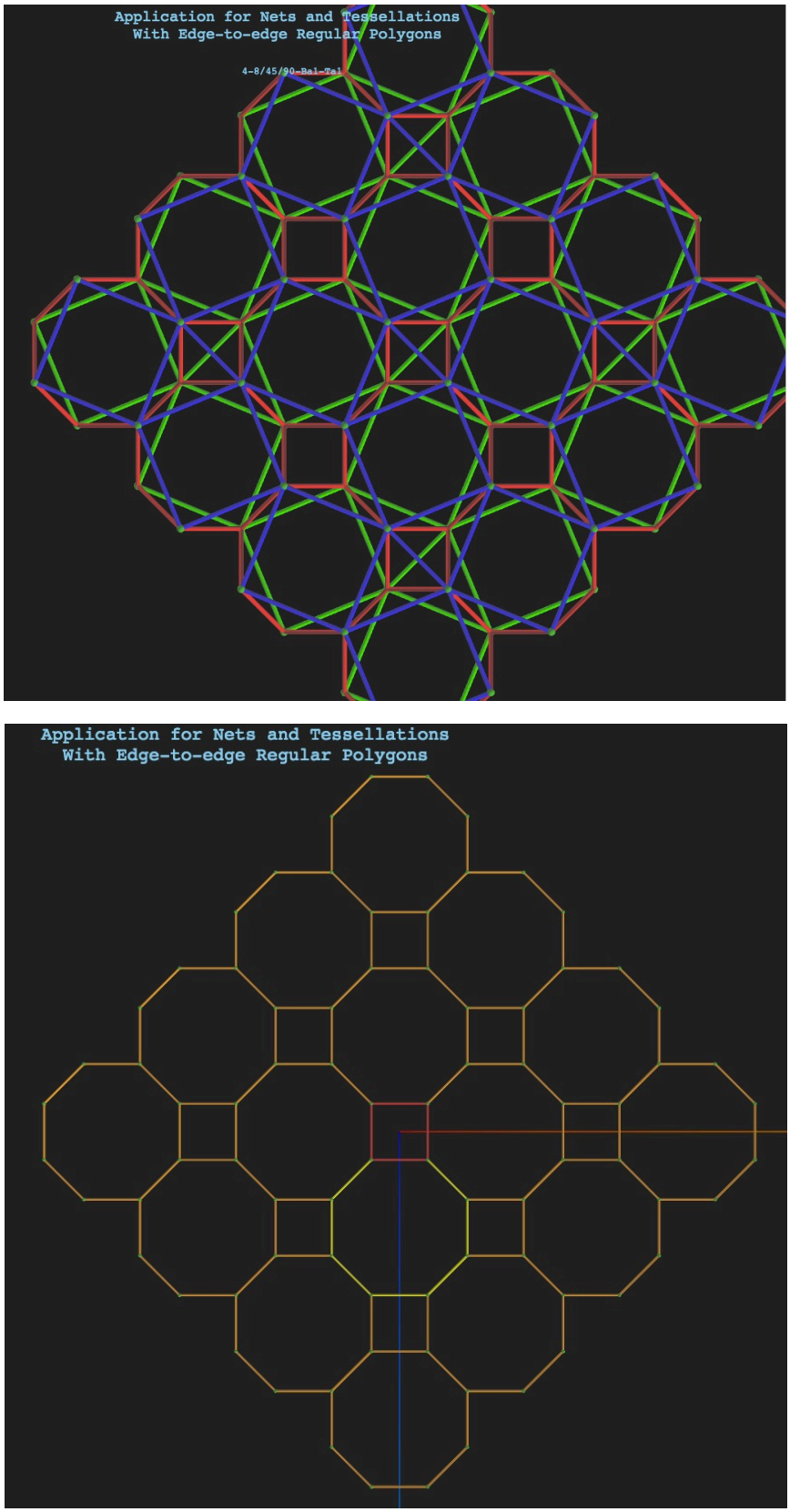


Figura 23: DLG 4-8/45/90-Ba1-Ta1 y el mosaico que lo origina.

Al presionar el botón “View Elements Properties”, ANTWERP nos indicará la colocación del polígono semilla, el número de fases, los polígonos que componen cada fase, y el tipo de simetrías. (Figura 24).

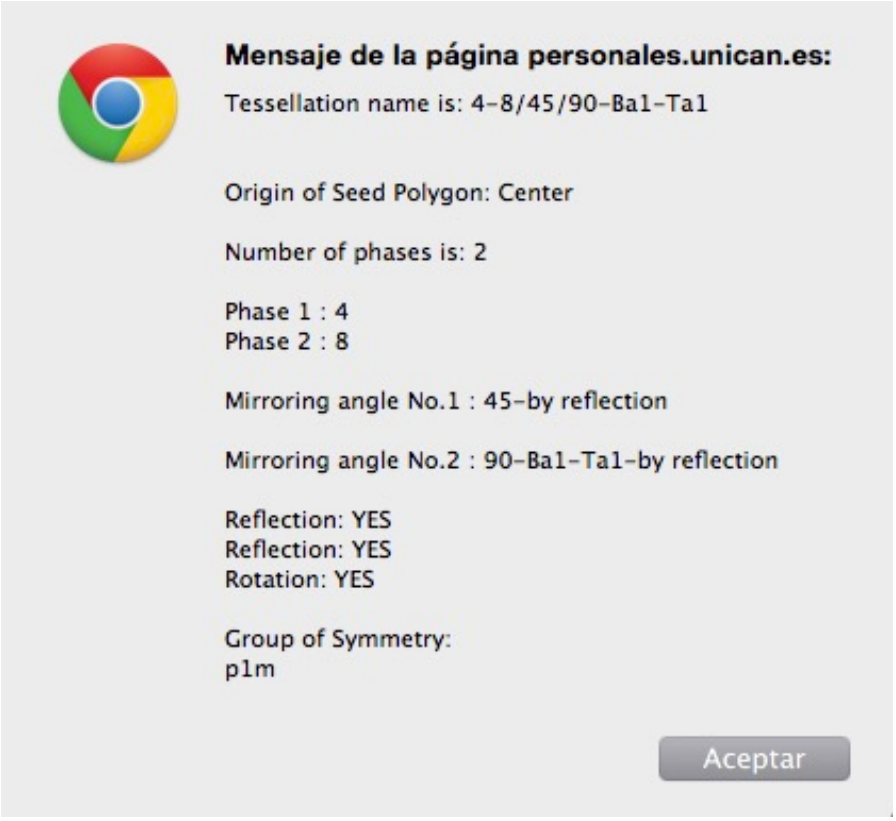


Figura 24: Propiedades de la DLG 4-8/45/90-Ba1-Ta1.

Procedemos ahora a exportar los datos de nodos y barras generados por ANTWERP en un preformateado ToyGL, generando un fichero .txt denominado “4-8-45-90-Ba1-Ta1 ToyGL.txt”. Dicho fichero quedará según se observa parcialmente en la Figura 25.

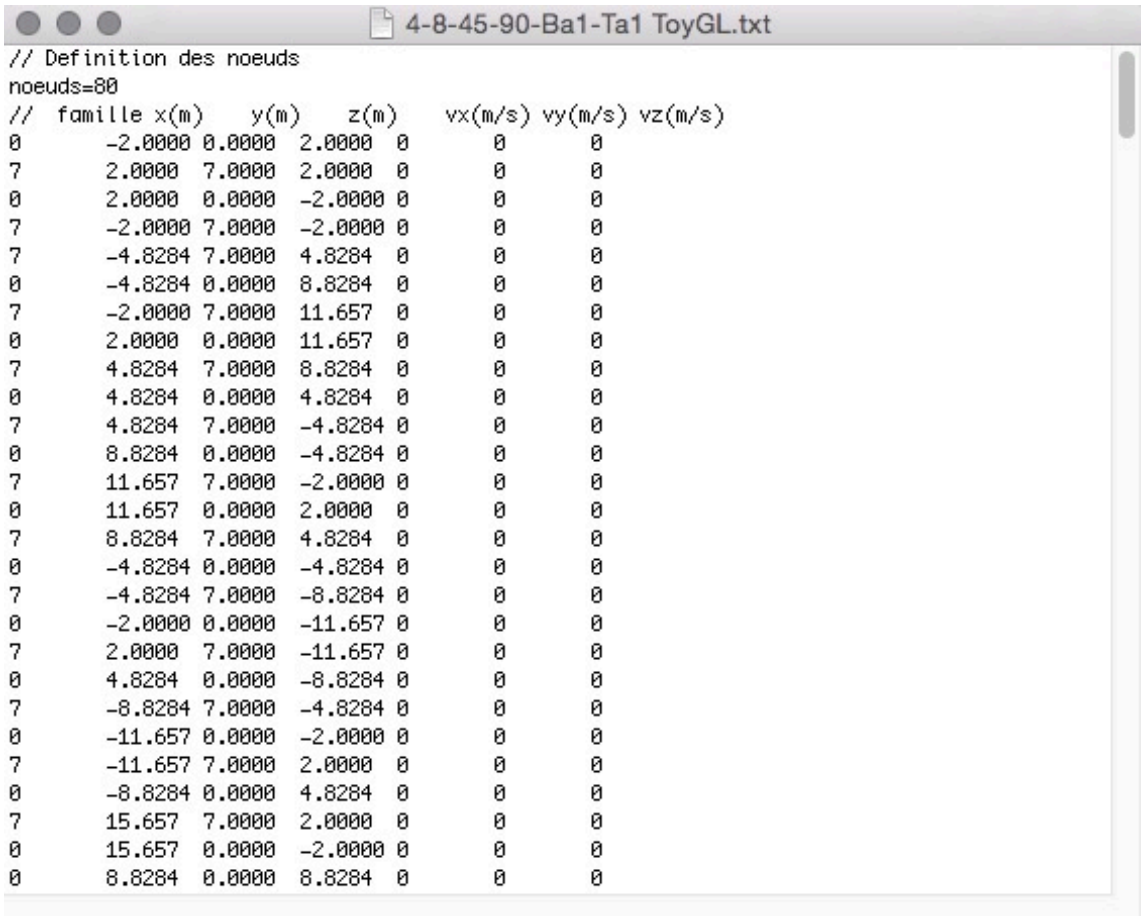


Figura 25: Información de los elementos del DLG para ToyGL.

Al iniciar la aplicación ToyGL y seleccionar el archivo generado por ANTWERP, se visualizarán los elementos al igual que se visualizarían en ANTWERP. Tras fijar los cuatro vértices extremos, asumiendo que serán los soportes de la estructura, y someter la malla a la acción gravitatoria, observaremos como la DLG es estable, no colapsa, y tienen un reparto relativamente homogéneo de tensiones, no alcanzando valores extremos. (Figuras 26 y 27).

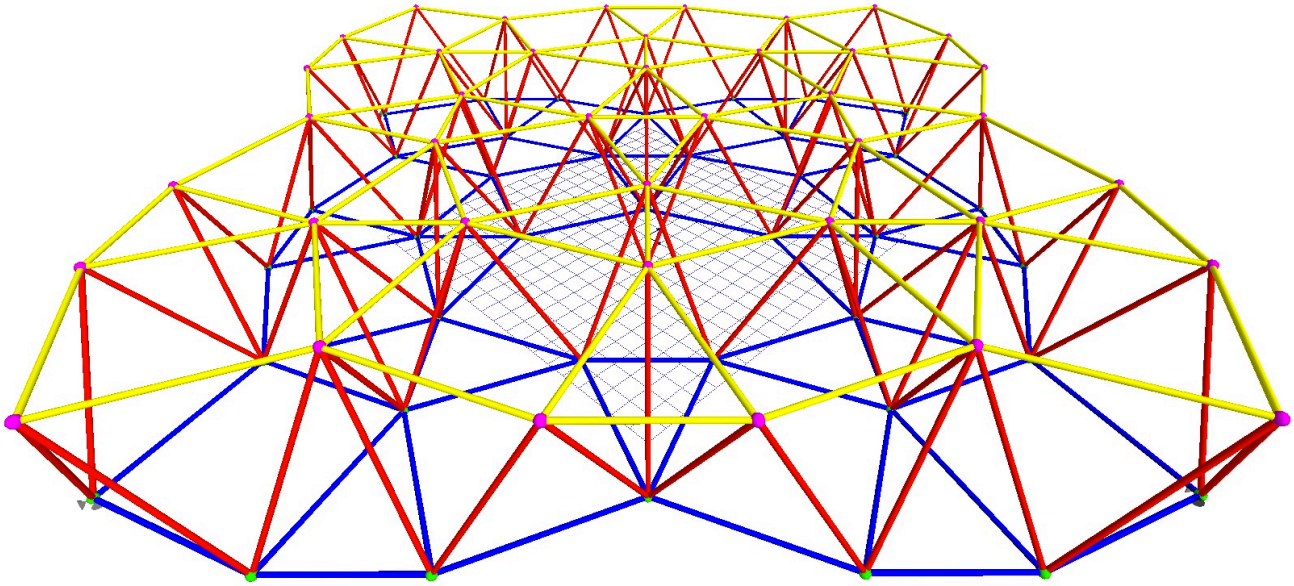


Figura 26: Representación de la DLG exportada, y cargada en ToyGL

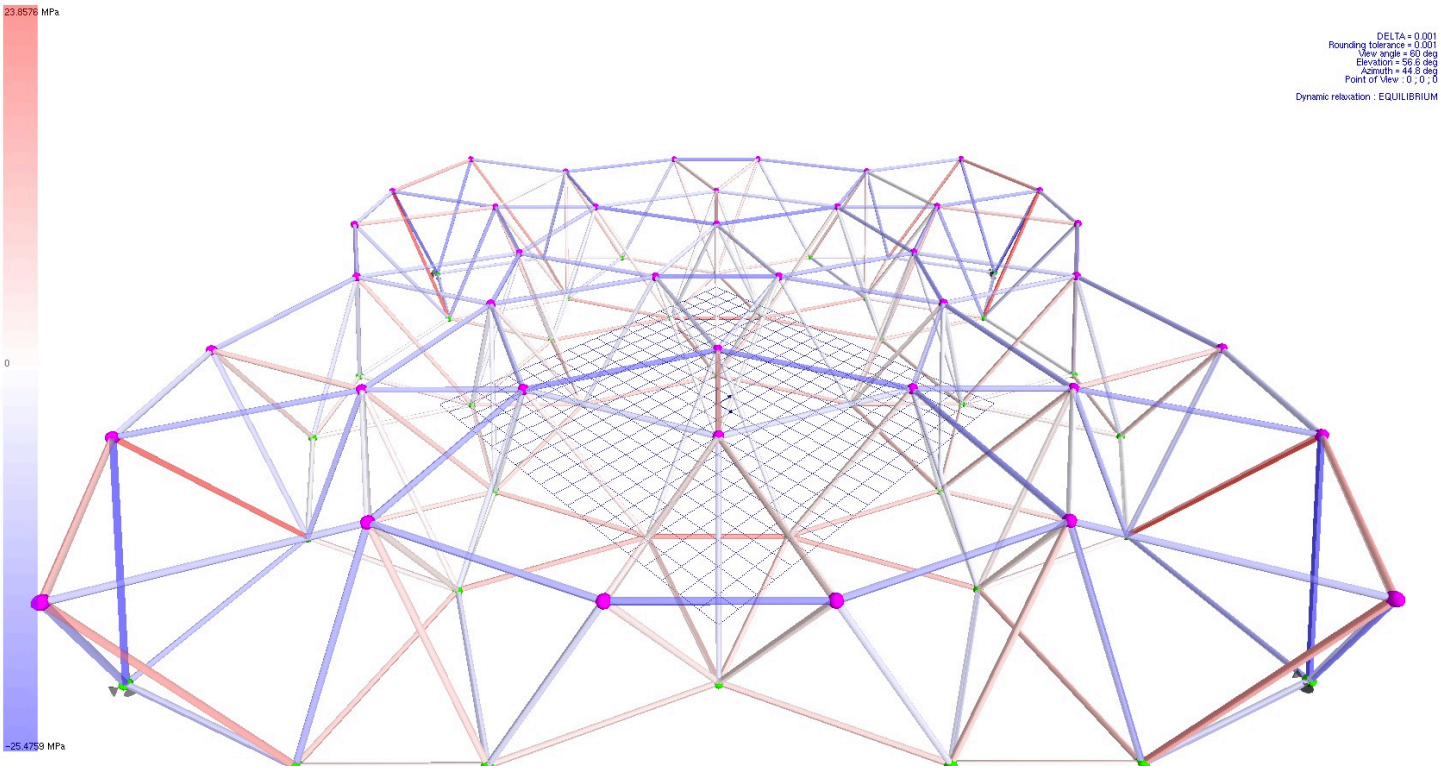


Figura 27: Representación del estado tensional de la DLG en ToyGL, bajo acción gravitatoria.

Los valores empleados (determinados por defecto en ToyGL y modificables), son los indicados en la siguiente tabla.

MATERIAL	ACERO
DENSIDAD	7850 Kg/m^3
E .TRACCIÓN	210 GPa
E. COMPRESIÓN	210 GPa
SECCIÓN	10.0833 cm^2

Como podemos observar en la tabla, el módulo de Young a tracción y a compresión es el mismo; estamos utilizando barras en toda la estructura, tanto en elementos a tracción como a compresión. Analizando los resultados, podríamos sustituir los elementos traccionados por cables, reduciendo el peso total de la estructura y por consiguiente, los esfuerzos.

La deflexión máxima es de 7.69 cm. La longitud de las barras no es la misma en toda la estructura; depende de la capa y posición de las mismas. La longitud mínima de barra en toda la estructura es de 5.65m y la más larga, de 8.06m .

Las tensiones máximas son 23.85 MPa y -25.48 MPa; las fuerzas máximas son 24.05 KN y -25.68 KN.

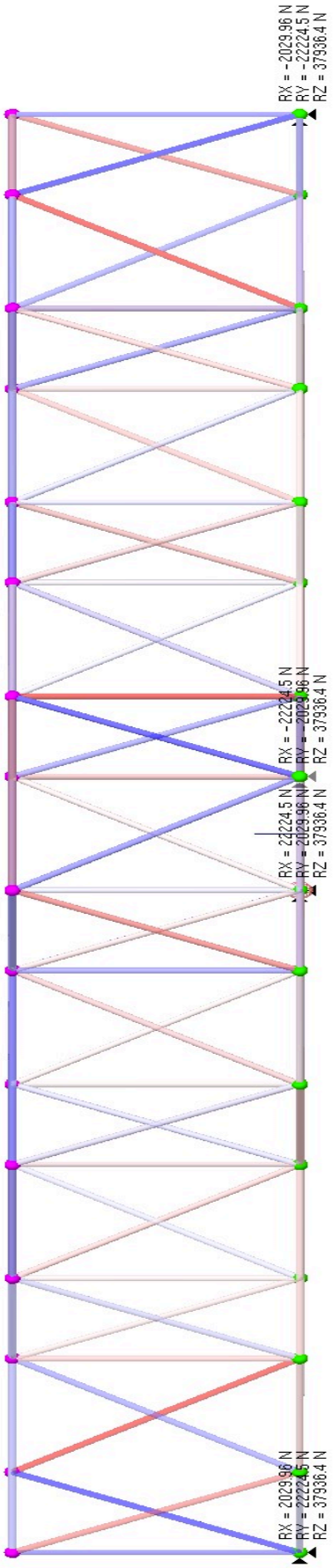


Figura 28: Representación de la malla y las reacciones en sus apoyos en perspectiva ortogonal.

La deformación de la estructura es inapreciable, al ser valores muy pequeños en comparación con las dimensiones totales.

7. CONCLUSIONES Y TRABAJOS FUTUROS

En este último capítulo de la memoria del proyecto se incluyen las conclusiones obtenidas de su realización y se plantean nuevas líneas de trabajo a través de mejoras o adaptaciones sobre el software.

ANTWERP abre nuevas líneas de investigación importantes, relacionadas con la búsqueda de un catálogo de mosaicos y DLG. Además, la posibilidad de exportar los diseños en un formato compatible con ToyGL y software CAD abre las puertas a un interesante número de aplicaciones de gran atractivo en diferentes áreas como la geometría y matemáticas, ingeniería mecánica y civil, arquitectura…

ANTWERP (Diaz-Gomez et al. 2015) se encuentra disponible en http://personales.unican.es/gomezjv/antwerp/ANTWERP_app.html

7.1- Objetivos alcanzados

Las metas alcanzadas satisfactoriamente han sido:

- La demostración de que mediante la nomenclatura propuesta para generar DLG y mosaicos se puede alcanzar un resultado inequívoco, correcto y consistente.
- La elaboración de un algoritmo que permita visualizar rápidamente cualquier malla y mosaico, no sólo un rango limitado y catalogado de ellas.
- Que funcione como aplicación web, sin necesidad de instalar software adicional.
- El diseño y la creación de una interfaz sencilla e intuitiva que facilite su accesibilidad para la mayoría de usuarios. Por otra parte, la usabilidad en dispositivos móviles necesita ser más trabajada en

próximas versiones para que pueda funcionar bien con la tecnología que ofrecen los dispositivos móviles, creando una web “responsive”.

- Presentar este trabajo como un artículo en el “Annual International Symposium on Future Visions” (Amsterdam, 17-21 Agosto 2015), organizado por la “International Association for Shell and Spatial Structures” (IASS) y la “Royal Netherlands Society for Engineers” (KIVI).

Este trabajo ha aportado además un cierto valor añadido a mi formación: la necesidad de aprender desde cero un nuevo lenguaje de programación y la participación en un proyecto de investigación. Como conclusión final, cabe destacar la satisfacción que supone a nivel personal haber conseguido completar todos los objetivos propuestos al comienzo del proyecto.

7.2- Trabajos futuros

7.2.1 Mejoras en la aplicación.

Sería interesante en próximas versiones mejorar la velocidad de representación de las mallas y la limpieza de memoria en la aplicación; incrementar el rendimiento en general para obtener una aplicación más fluida, especialmente en dispositivos móviles.

7.2.2 Búsqueda de nuevos mosaicos.

Al desarrollar ANTWERP una nueva idea surgió: usar un método estocástico que probase infinidad de posibilidades en varios minutos dentro de un bucle computacional, validando los resultados y descartando los mosaicos que no

rellenasen completamente el espacio o que creasen un solapamiento poligonal. ¿Sería posible encontrar nuevos mosaicos usando algoritmos genéticos o evolutivos? ¿Sería posible crear un inmenso catálogo con la totalidad de los mosaicos 2, 3, y n-uniformes?

Además sería interesante estudiar nuevas posibles reglas en la nomenclatura, que lleven a permitir representar un mayor rango de mosaicos y mallas.

7.2.3 Búsqueda de nuevas mallas de doble capa.

Inspirado por la idea anterior, otra posible línea de investigación podría ser desarrollar un nuevo algoritmo que obtuviese nuevos diseños de DLG a partir de cualquier tipo de mosaico. A partir de este proceso de prueba y error, varios nuevos ejemplos de DLG se podrían encontrar y aplicar a ellos las leyes de Otero reformuladas.

7.2.4 Búsqueda de nuevas mallas de doble capa tensegríticas.

Las mallas de doble capa tensegríticas (DLTG) se diferencian de las DLG tal que, en las DLTG, los elementos están dispuestos según una configuración que hace que los elementos bajo compresión (barras o vigas) estén aislados y no estén en contacto entre ellos; se encuentran conectados y estabilizados mediante elementos bajo tracción (cables y tendones). Recientemente un nuevo método fue propuesto (Gómez-Jáuregui et. al. ,2012), para “tensegrizar” DLG convencionales. Sería muy interesante introducir esta serie de operaciones, denominadas “manipulaciones de Rot-Umbela”, en próximas versiones.

BIBLIOGRAFÍA Y REFERENCIAS

- Averseng, J., Quirant, J. & Dubé, J.-F., Interactive design and dynamic analysis of tensegrity systems,). International Journal of Space Structures, vol. 27, no. Special Issue 2–3, 2012.
- Berners-Lee, World Wide Web Consortium 1991. http://www.w3.org/
- Critchlow, K., Order in space: a design source book. London: Thames and Hudson, 1969.
- Cundy, H. M. and Rollett, A. P., Mathematical models. Stradbroke: Tarquin, 1981.
- Diaz-Gomez, S., Gomez-Jauregui, V., Manchado, C., and Otero, C., ANTWERP v2.0 [software]. www.tensegridad.es/antwerp, 2015.
- Flanagan, D. JavaScript: The Definitive Guide. (6th Edition). Editorial O`Reilly. Última consulta 2015.
- Fog Creek Software 2010, Trello . https://trello.com/about
- Gomez-Jauregui, V., ANTWERP v1.0 [software]. http://www.tensegridad.es/DLG- tessellations/DLG-tessellations.html, 2012.
- Gomez-Jauregui, V., Arias, R., Otero, C., and Manchado, C., Novel Technique for Obtaining Double-Layer Tensegrity Grids. International Journal of Space Structures, 2012; 27 (Special Issue 2–3); 155–166.
- Gomez-Jauregui, V., Otero, C., Arias, R., and Manchado, C., Generation and Nomenclature of Tessellations and Double-Layer Grids. Journal of Structural Engineering ASCE, 2012; 138 (7); 843–852.
- Malla, R. B. and Serrette, R. L., Double-layer grids: review of static and thermal analysis methods. Journal of Structural Engineering ASCE, 1996; 122 (8); 873–881.
- Microsoft Visual Studio webpage. https://www.visualstudio.com/es-es/visual-studio-homepage-vs.aspx
- Otero, C., Diseño geométrico de cúpulas no esféricas aproximadas por mallas triangulares con un número mínimo de longitudes de barra, PhD Thesis, Universidad de Cantabria, Santander, 1990.