

Desarrollo de una Interfaz Gráfica para una Herramienta de Cálculo de Estructuras

Rafael Olivera - Federico García

29 de noviembre de 2015

Índice

1. Introducción	6
1.1. Definición del problema y motivación	6
1.2. Desarrollo previo	7
1.3. Objetivos y resultados esperados	8
1.4. Desarrollo del proyecto	9
1.5. Organización del documento	9
2. Estado del Arte	10
2.1. Cálculo de estructuras	10
2.1.1. Cálculo implicados	10
2.1.2. IETFEM	10
2.2. Herramientas comerciales	10
2.2.1. SAP2000	11
2.2.2. AxisVM	11
2.2.3. Herramientas Web	11
2.3. Desarrollo 3D	12
2.3.1. OpenGL	12
2.3.2. Java 3D	12
2.3.3. WebGL	12
2.3.4. Otras herramientas	12
2.4. Desarrollo 3D en la Web	12
2.4.1. HTML5 - Canvas	12
2.4.2. Librerías para desarrollo 3D	12
2.4.3. Interacción con el usuario	13
2.5. Información complementaria	13
2.5.1. Investigación sobre proyectos similares en América Latina	13
2.5.2. Herramientas de cálculo de estructuras en la Web	13
3. Organización del trabajo	13
3.1. Alcance	13
3.2. Metodología de trabajo	15
3.3. Estimación y esfuerzo efectivo	16
4. Presentación de la solución	18
4.1. Análisis y relevamiento de requerimientos	18
4.2. Diseño de la solución	20
4.2.1. Decisiones tomadas	20
4.2.2. Diseño final	22
4.3. Arquitectura	23
4.4. Tecnologías y herramientas utilizadas	23
4.4.1. HTML5 - Javascript - CSS3	23

4.4.2.	Bootstrap	23
4.4.3.	AngularJS	23
4.4.4.	ThreeJS	23
4.4.5.	Electron	23
4.5.	Manejo del espacio 3D	24
4.5.1.	Eventos de usuario	24
4.5.2.	Adición, sustracción y transformación de objetos	24
4.5.3.	Manejo de la cámara	24
4.5.4.	Trazado de rayos e intersecciones con objetos	24
4.5.5.	Performance	24
4.6.	Manejo de datos	25
4.6.1.	Entrada de información (dibujado e importación)	25
4.6.2.	Mantenimiento de la estructura durante el proceso de dibujo	25
4.6.3.	Almacenamiento de la estructura	25
4.6.4.	Salida de Datos	25
4.7.	Análisis de resultados del Core	26
4.7.1.	Generación de resultados	26
4.7.2.	Introducción de datos en la UI	26
4.7.3.	Visualización	26
5.	Resultados obtenidos	26
5.1.	Comparación IETFEEM con y sin UI	26
5.1.1.	Análisis del impacto en la usabilidad	26
5.1.2.	Análisis del impacto en el tiempo de ejecución	26
5.2.	Casos de prueba	27
5.2.1.	Estudio de casos de pequeño porte (Torre pequeña)	27
5.2.2.	Estudio de casos de mediano porte (Grúa)	27
5.2.3.	Estudio de casos de gran porte y pruebas de stress (Torre Eiffel)	27
6.	Conclusiones y trabajo futuro	27
6.1.	Conclusiones	27
6.2.	Trabajo a futuro	28
6.2.1.	Trabajo en el motor	28
6.2.2.	Trabajo en la interfaz	28
6.2.3.	Despliegue de la aplicación	28
7.	Anexos	28

1. Introducción

1.1. Definición del problema y motivación

Desde las primeras casas construidas por el hombre, hasta el edificio más moderno y extravagante que exista en la actualidad, puede decirse que se buscó en el fondo el mismo objetivo: lograr una estructura segura, resistente y funcional. Hoy por hoy, la evolución del conocimiento humano y de la tecnología circundante ha permitido desarrollar a niveles altísimos el comprendimiento del problema y sus posibles soluciones.

El cálculo de estructuras, en ese sentido, es una rama fundamental dentro de la ingeniería civil. Se trata de una serie de complejos cálculos realizados con la finalidad de lograr estructuras óptimas con las condiciones descritas anteriormente. A grandes rasgos, se busca que la estructura pueda soportar tanto su propio peso, como cualquier fuerza externa que pueda ser aplicada a la misma, lo que puede derivar en que la misma no siempre sea igual en la realidad a como se diseñó. Esto quiere decir que la estructura puede sufrir ciertas deformaciones antes de alcanzar su punto de equilibrio.

La Ingeniería en Computación no ha dejado este problema de lado, ya que existen diversos sistemas informáticos encargados de facilitar el diseño y cálculo de estructuras. Estos sistemas permiten, a grandes rasgos, dibujar una estructura mediante la definición de diferentes materiales, secciones, fuerzas externas, etc. Finalmente, realizan los cálculos correspondientes, mostrando la estructura en un estado de equilibrio y las deformaciones ocurridas en el proceso.

Así como existen estos sistemas reconocidos mundialmente, la Facultad de Ingeniería cuenta también con su propio sistema de cálculo de estructuras. Su nombre es IETFEM, y fué desarrollado por los Ing. Pablo Castrillo y Jorge Pérez pertenecientes al Instituto de Estructuras y Transporte(IET). Se trata de un motor de cálculo desarrollado en Octave que recibe una estructura descrita en formato texto y genera gráficas e imágenes con la deformación de la misma.

En este proyecto, se realizará una interfaz gráfica acorde para ser utilizada en conjunto con el motor de cálculo antes mencionado, y lograr así un sistema completo de diseño y cálculo de estructuras. Se busca, en particular, agregar funciones de dibujado y visualización de resultados que pueden observarse en otros sistemas de la misma índole, acercando al IETFEM a los sistemas comerciales y logrando una mayor usabilidad y eficiencia para los estudiantes que lo utilizan.

36 1.2. Desarrollo previo

37 Como mencionamos anteriormente, la FING cuenta con un motor de cálculo
38 de estructuras desarrollado por los Ingenieros Pablo Castrillo y Jorge Pérez. El
39 mismo resuelve problemas de cálculo de estructuras utilizando el Método de
40 Elementos Finitos(MEF).

41 El MEF es, desde mediados del siglo XX, una de las principales herramientas
42 utilizadas por los ingenieros para el análisis de sistemas estructurales, mecáni-
43 cos, eléctricos, etc. El avance de la computación y la disponibilidad creciente de
44 computadores potentes a bajo costo ha provocado que los programas comercia-
45 les de MEF para el cálculo estructural sean utilizados masivamente. De hecho,
46 en los últimos cuarenta años el MEF ha transformado los procedimientos de tra-
47 bajo de todas las áreas de ingeniería y constituye hoy una de las herramientas
48 indispensables con las que un ingeniero debe contar en el ejercicio de su profe-
49 sión. Por otra parte, el uso del MEF por parte de profesionales no debidamente
50 capacitados podría eventualmente producir errores en el diseño de estructuras,
51 y por tanto, riesgos para los usuarios.

52 En este contexto, la enseñanza del MEF en las carreras de Ingeniería se trans-
53 forma en un desafío docente, donde además de formar a los estudiantes en el
54 uso de diferentes programas de cálculo estructural es necesario transmitirles los
55 conocimientos y herramientas que les permitan realizar un análisis crítico de los
56 resultados. Es importante destacar además, que la mayoría de los programas
57 comerciales (ej: SAP2000 y AxisVM) de MEF son de código cerrado, por lo que
58 presentan como desventaja a nivel educativo, que no permiten a los estudiantes
59 ver su funcionamiento interno, limitando la comprensión de los errores durante
60 el aprendizaje.

61 De esta manera surge entre docentes del Grupo de Mecánica de Sólidos Comp-
62 putacional (MSC) del Instituto de Estructuras y Transporte (IET) la motivación
63 de brindar una solución al problema a través del desarrollo de un software edu-
64 cativo adecuado: IETFEM.

65 IETFEM comenzó a desarrollarse en 2012 . El primer módulo desarrollado per-
66 mitió resolver problemas de estructuras de barras articuladas o apoyadas en
67 el plano con cargas aplicadas en los nodos. Esta primera versión fue utilizada
68 por estudiantes del curso de Elasticidad 2013; luego se incluyó la posibilidad de
69 generar un informe de salida en formato LaTeX. Posteriormente, la herramien-
70 ta contó con el aporte del docente del IET, Agustín Spalvier, desarrollando la
71 capacidad de ingresar cargas distribuidas uniformes en elementos de pórtico y
72 el análisis modal de vibraciones de pórticos. Finalmente, a principios de 2014,
73 Castrillo desarrolló un módulo para la resolución de problemas con variaciones
74 de temperatura y fuerza de volumen en barras articuladas.

75 Se buscó una herramienta que sin ser compleja para su aplicación en cursos
76 de grado, permita al estudiante visualizar el funcionamiento interno del méto-
77 do de cálculo. Por ello se optó por la sintaxis de programación de GNUOctave

78 (herramienta libre de alta compatibilidad con Matlab), ya conocida por los es-
79 tudiantes. Se considera que contar con un software abierto donde los estudiantes
80 pueden entender e incluso programar nuevos cálculos de acuerdo a sus necesi-
81 dades, enriquece el trabajo desde el punto de vista didáctico.

82 La forma de ingreso de datos se eligió de acuerdo a otros programas de cálculo
83 de estructuras como SAP2000 donde se deben definir: materiales, secciones,
84 estados de carga, geometrías, conectividades, etc. En el IETFEM se optó por
85 una entrada de archivo de texto plano donde el estudiante debe ingresar esta
86 información. La salida también es en texto plano (.txt y .tex) y gráfica, al igual
87 que en los programas comerciales.

88 Sin embargo, la generación del archivo de entrada y la comunicación con el
89 IETFEM pueden llegar a ser tediosas y complicadas para el estudiante. Debe
90 tenerse en cuenta que debe especificarse la estructura nodo por nodo, barra
91 por barra, describiendo los materiales, secciones y fuerzas aplicadas, entre otras
92 cosas, respetando además un formato fijo de documento que puede derivar en
93 diversos errores de sintaxis.

94 Por lo tanto, se desarrollará en este proyecto una interfaz gráfica de código
95 abierto donde el estudiante pueda dibujar la estructura de una manera sencilla
96 e intuitiva, y que genere la entrada al IETFEM de manera automática. De esta
97 manera, se pretende mejorar tanto la facilidad de uso como la eficiencia del
98 mismo.

99 1.3. Objetivos y resultados esperados

100 Como mencionamos antes, a pesar de la increíble potencia en la resolución del
101 problema del cálculo de estructuras, IETFEM presenta ciertos puntos a mejorar
102 para ser comparado con otros sistemas del mismo rubro.

103 A lo largo de este proyecto perseguimos 2 grandes objetivos que consideramos
104 esenciales para el enriquecimiento del sistema: Mejorar la eficiencia y mejorar
105 la usabilidad

106 Para mejorar la usabilidad, se desarrollará una interfaz que permita al usuario
107 dibujar la estructura de manera fluida y amigable. Se trata de un espacio 3D
108 donde el usuario puede moverse libremente utilizando el mouse para mover y
109 rotar la cámara. Se podrá dibujar la estructura de una manera continua e in-
110 intuitiva. Además, facilitará la comunicación con el motor de cálculo previamente
111 desarrollado y la visualización de los resultados obtenidos.

112 Para mejorar la eficiencia, reduciremos el tiempo de ejecución del motor de
113 cálculo, eliminando el proceso de graficación y generación de imágenes, ya que
114 ahora los resultados podrán verse en la nueva interfaz. Como regla básica, bus-
115 camos que el usuario pierda el menor tiempo posible en problemas tecnológicos
116 o informáticos y que dirija sus esfuerzos al comprendimiento del problema en sí
117 y su método de resolución.

118 A modo de resumen, se busca realizar un sistema ágil, de código abierto, que
119 mejore ambos aspectos lo suficientemente como para poder ser utilizado sin pro-
120 blemas en el curso de Elasticidad dictado por el IET. Con el fin de verificar el
121 cumplimiento de los objetivos por parte del sistema, una vez finalizado, será
122 evaluado resolviendo ejercicios del curso antes mencionado, realizando compa-
123 raciones y análisis del tiempo de ejecución.

124 1.4. Desarrollo del proyecto

125 El proyecto comenzó con una fase fuerte de investigación. Inicialmente se reali-
126 zaron reuniones ocasionales con los clientes, donde se reunió información valiosa
127 sobre el problema de cálculo de estructuras y el método de elementos finitos.
128 Además se definió qué tipo de herramienta se quería, qué funcionalidades eran
129 deseadas y qué objetivos se buscaban. Durante esta etapa se utilizó el motor
130 de cálculo directamente para comprender su funcionamiento y compararlo con
131 otras herramientas comerciales.

132 Una vez comprendido el problema, se procedió a buscar herramientas con las
133 cuáles desarrollar la interfaz. Se investigaron librerías y lenguajes de progama-
134 ción 3D, optando al final por utilizar tecnologías web por su simplicidad de uso,
135 agilidad y portabilidad

136 Posteriormente se comenzó a diseñar e implementar la herramienta, separan-
137 do en diferentes módulos que serán descriptos en detalle en el capítulo 4. Se
138 ejecutaron reuniones quincenales con los clientes para definir detalles, corregir
139 errores, evaluar resultados y tomar decisiones en conjunto. Esta fase ocupó la
140 mayor parte del tiempo del proyecto, debido a la dificultad técnica del mismo.

141 Finalmente, una vez alcanzado un producto inicial que cumplía las expectativas
142 planteadas, se procedió a realizar pruebas sobre el mismo, detectando ciertos
143 errores de performance que fueron solucionados hasta un nivel considerable-
144 mente bueno(se hablará de estas medidas en el capítulo 5).

145 1.5. Organización del documento

146 El resto del documento se organiza de la siguiente manera:

147 En el siguiente capítulo comenzamos analizando el estado del arte, tanto del
148 problema de cálculo de estructuras como de herramientas de programación 3D,
149 y su posible uso en sistemas de este tipo. Se realiza un estudio de diferentes
150 herramientas investigadas, el estado de las mismas y su posibilidad de ser uti-
151 lizadas en este proyecto. También se investigan otros sistemas de cálculo de
152 estructuras y otro proyectos académicos similares en América Latina.

153 Posteriormente, en el capítulo 3, hablaremos de la organización del trabajo a
154 lo largo del proyecto. Hablaremos del alcance del mismo, definiendo las fun-

155 cionalidades y características específicas que se buscan en el producto final. Se
156 describirá la metodología de trabajo utilizada y se realizarán estimaciones para
157 cada tarea comprendida, comparando finalmente con el esfuerzo efectivo.

158 A continuación, en el capítulo 4, se procederá a plantear la solución propuesta,
159 detallando cada aspecto de la misma. Se describirá con exactitud su proceso
160 de diseño e implementación, la arquitectura definida, el funcionamiento de cada
161 componente, las herramientas utilizadas y su uso en general.

162 En el capítulo 5, se especifican los resultados obtenidos, analizando diferentes
163 casos de prueba y comparando con resultados obtenidos desde IETFEM antes
164 de la realización de este proyecto. Se analizan además los problemas obtenidos
165 durante esta fase y cómo fueron resueltos.

166 Finalmente, el 6 capítulo, enumera las conclusiones obtenidas durante el pro-
167 yecto, analizando el cumplimiento de objetivos y proponiendo posible trabajo a
168 futuro a desarrollar sobre IETFEM.

169 **2. Estado del Arte**

170 **2.1. Cálculo de estructuras**

171 **2.1.1. Cálculo implicados**

172 aca hay que hablar el problea, les mandamos un mail paraver donde podemos
173 leer mas

174 **2.1.2. IETFEM**

175 hablar de como resuelve el ietfem los problemas de arriba

176 beneficios del ietfem, resuelve probelmas complejos como los comerciales

177 carencias del ietfem, mencionar lso problemas un pocomas en detalle, mas que
178 nada haciendo referencia a que a pesar de que soluciona probleas complejos
179 eficientemente, asi como esta es muy dificil de usar

180 **2.2. Herramientas comerciales**

181 Existen en el mercado diversos productos de software enfocados al análisis de
182 estructuras, con gran cantidad de funcionalidades y utilizados por ingenieros de
183 todo el mundo en problemas reales. En el marco de este proyecto se exploraron
184 con mayor rigurosidad dos herramientas: SAP2000 y AxisVM, las cuales en
185 etapas mas avanzadas del desarrollo fueron tomadas como estándar para la
186 implementación de ciertas funcionalidades.

187 **2.2.1. SAP2000**

188 Es un software comercial desarrollado por la empresa Computers & Structures, Inc. fundada en 1975 en California, siendo uno de los pioneros en herramientas de análisis de estructuras.

191 Actualmente en su versión 18, SAP2000 es una aplicación para computadoras que se ejecuta en ambientes Windows. Cuenta con un entorno gráfico 3D para el modelado y una interfaz de usuario muy completa que puede resultar demasiado compleja en los primeros pasos.

195 Entre las características mas importantes se encuentran:

- 196 ■ Un motor de análisis que puede resolver varios tipos de problemas tales como: (LES PREGUNTAMOS A LOS PROFES XQ ES UN HUEVO DE ENTEDNER),
- 199 ■ Diversas características para el modelado como templates, sistema de grillas, distintas vistas y herramientas de meshing.
- 201 ■ Diversos componentes estructurales como articulaciones, barras, cables sólidos, resortes, etc.
- 203 ■ Posibilidad de aplicar distintos tipos de cargas.
- 204 ■ Varias posibilidades para ver la salida de los cálculos con diagramas, tablas y reportes.
- 206 ■ Importación y exportación de modelos en distintos formatos estándar.

207 Por todo esto SAP2000 es uno de los productos líderes en el mercado siendo utilizado en mas de 160 países en todo el mundo.

209 **2.2.2. AxisVM**

210 Es un software comercial desarrollado por la empresa InterCAD Kft. en 1991 y con sede en Hungría. Fue una de las primeras herramientas 3D basada en el método de los elementos finitos.

213 Actualmente en su versión 13, AxisVM requiere computadoras con el sistema operativo Windows. Cuenta con un entorno gráfico 3D para el modelado y una interfaz de usuario muy completa que puede resultar demasiado compleja en los primeros pasos.

217 **2.2.3. Herramientas Web**

218 hablar de cada uno, quien lo desarrolla, si se puede en que esta hecho

219 hablar del impacto mundial, que problemas resuelve, hablar de las cosas que nos gustaria poner en nuestro programa, cual nos gustaria evitar o mejorar

221 **2.3. Desarrollo 3D**

222 **2.3.1. OpenGL**

223 investigar y hablar de open gl, que es, en que se usa, etc...

224 hablar de porque no la elegimos, dificultad de uso, poca experiencia, etc...

225 **2.3.2. Java 3D**

226 lo mismo que al anterior

227 **2.3.3. WebGL**

228 lo mismo que el anterior

229 hablar ademas de que al descubrir esta opcion se nos desperto la idea de hacerlo
230 eb

231 **2.3.4. Otras herramientas**

232 hablar de otras herramientas de escritorio que hayamos visto, no se que mas
233 poner aca

234 **2.4. Desarrollo 3D en la Web**

235 **2.4.1. HTML5 - Canvas**

236 que es html5

237 Hablar de las facilidades que da el html5 para cosas 3d mediante el canvas

238 posibles conexiones entre ebgl y canvas

239 **2.4.2. Librerías para desarrollo 3D**

240 hablar de las libreirias que vimos sobre ebgl, threejs y las otras que estuvios
241 viendo que no me acuerdo

242 porque elegimos threejs y que beneficios se tienen

243 hacer un analisis si se pueden cumplir los objetivos con esta tecnologia

244 **2.4.3. Interacción con el usuario**

245 como pretendemos que sea la interaccion con el usuario, basandonos en los
246 programas comerciales y lo que ofrece la eb
247 plantearse cambiar el titulo de esta seccion para abarcar mas contenido

248 **2.5. Información complementaria**

249 **2.5.1. Investigación sobre proyectos similares en América Latina**

250 buscar proyectos similares en internet, en america latina y el mundo, y compa-
251 rarlos
252 comparar lo encontrado con ietfem, y rematar señalando que es el primer pro-
253 yecto de este tipo en sudamerica

254 **2.5.2. Herramientas de cálculo de estructuras en la Web**

255 Investigar si existen
256 en casod e que existan, hacer una mini comparacion con lo que seria ietfem eb
257 mencionar que no existan muchas herramientas y que tendria mucho potencial

258 **3. Organización del trabajo**

259 **3.1. Alcance**

260 Como mencionamos anteriormente, los objetivos planteadas en este proyecto se
261 basan en mejorar tanto la eficiencia como la usabilidad del IETFEM. En ese
262 sentido, existen dentro de la rama del cálculo de estructuras una infinidad de
263 funcionalidades y mejoras posibles que pueden resultar útiles en nuestro sistema.
264 Por lo tanto, se definió un conjunto acotado de funcionalidades y característi-
265 cas deseables en el producto final, apuntando a alcanzar satisfactoriamente los
266 objetivos planteados y lograr una herramienta de alto nivel.

267 Se consideró como prioridad apuntar a una herramienta académica, es decir, una
268 herramienta libre, intuitiva para los estudiantes y aplicable en cursos dictados
269 por el IET. En particular, se tomó como referencia el curso de Elasticidad, curso
270 donde ya fué utilizado satisfactoriamente el IETFEM y donde será utilizado
271 luego de la realización de este proyecto.

272 La principal y más grande funcionalidad que se desarrollará será le presencia
273 de un espacio 3D. El mismo será el elemento central de la aplicación, mediante

274 la cual el usuario efectuará la mayor parte de las interacciones posibles. Se
275 pretenden integrar dentro de este espacio 3D las siguientes funcionalidades:

- 276 ■ Rotación de la cámara de visualización.
- 277 ■ Movimiento de la misma por todo el espacio 3D.
- 278 ■ Zoom In y Zoom Out.
- 279 ■ Dibujado de nodos y barras.
- 280 ■ Dibujado de grillas auxiliares.
- 281 ■ Selección de nodos y barras: Para setear propiedades en las mismas.
- 282 ■ Eliminación de nodos y barras.
- 283 ■ Visualización de propiedades: Fuerzas, puntos de apoyo y resortes.
- 284 ■ Visualización de estructura resultante: Observar la deformada y comparar
285 con estructura original.
- 286 ■ Escalamiento la estructura deformada: «Exagerar» la deformación, para
287 apreciar pequeñas deformaciones.
- 288 ■ Visualización de las propiedades de la estructura deformada utilizando
289 escalas de colores: Deformación, Fuerzas, Tensiones, etc.

290 Más allá de que se pretende que el usuario tenga una experiencia interactiva
291 mediante el dibujo de la estructura, es necesario definir ciertas funcionalidades
292 fuera del espacio 3D. Ya sea tanto por comodidad como por intuición, estas
293 opciones se encuentran en diferentes menús que rodean el espacio, similar a los
294 demás programas comerciales dentro del rubro que se investigaron en el capítulo
295 anterior.

- 296 ■ Abrir y Guardar Estructuras.
- 297 ■ Definición de Materiales.
- 298 ■ Definición de Secciones.
- 299 ■ Asignar propiedades a barras: Material y sección.
- 300 ■ Asignar propiedades a nodos: Fuerzas, Apoyos y Resortes.
- 301 ■ Selección de nodos y barras: Para setear propiedades en las mismas.
- 302 ■ Eliminación de nodos y barras.
- 303 ■ Prendido y apagado de elementos auxiliares
- 304 ■ Seteo de Factor de escalamiento para la estructura deformada

305 Si bien estos elementos nos permiten estimar una interfaz gráfica completa e
306 intuitiva, resta definir aún la funcionalidad más importante del proyecto: la
307 comunicación con el motor de cálculo. La salida de la interfaz debe ser un
308 archivo reconocible por el IETFEM, del cuál pueda obtener todos los datos de

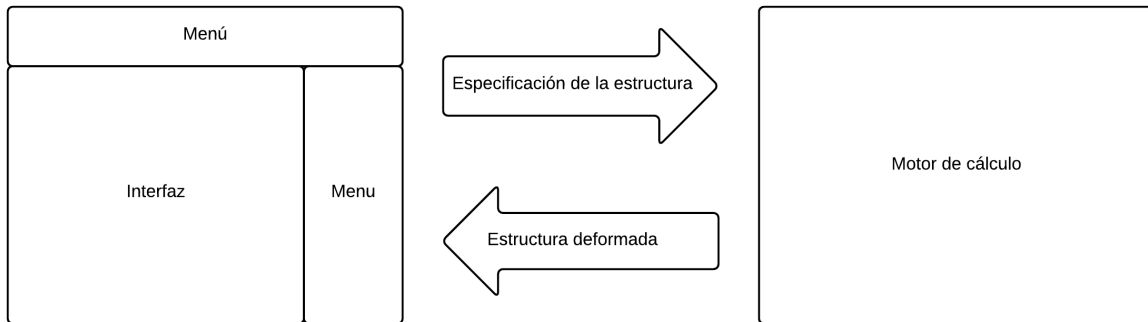


Figura 1: Ciclo de vida de IETFEM

la estructura. Así mismo, el motor debe ofrecer como salida otro archivo, el cuál será recibido por la interfaz con el fin de mostrar los resultados obtenidos. Dicha comunicación puede observarse en la Figura 1, aunque se hará hincapié en cómo se resolvió esta comunicación en el siguiente capítulo.

3.2. Metodología de trabajo

En las primeras etapas del proyecto se focalizó el trabajo en comprender el problema que se quiere resolver. Se tuvieron reuniones quincenales con los clientes dónde se habló del problema del cálculo de estructuras y cómo lo resuelve IETFEM. Dichas reuniones se apoyaron además en una permanente comunicación por e-Mail y una vasta investigación del problema por nuestra parte. Para esto no sólo se investigó sobre el problema, sino que además se utilizaron productos similares e incluso el propio IETFEM con ejemplos simples.

Una vez comprendido el problema, se pasó a buscar una solución al mismo. Dentro de esta etapa se pueden incluir la búsqueda de herramientas, el análisis y el diseño de la aplicación. Se mantuvieron las reuniones con los clientes, evaluando herramientas y enseñando prototipos realizados a modo de prueba. Se investigaron lenguajes y librerías de programación 3D, tanto de escritorio como web, decidiendo en última instancia utilizar WebGL (se hablará más en detalle en el siguiente capítulo).

Conforme pasaba el tiempo las reuniones se fueron enfocando cada vez más en el producto final, comenzando a definir las funcionalidades y características del mismo. Mientras se mantenía contacto con los clientes, se realizó por nuestra parte la definición de casos de uso, con sus respectivos diagramas de flujo, al tiempo que se definió la arquitectura del sistema en base a los requerimientos obtenidos y las prestaciones de las herramientas definidas.

Finalmente, para las etapas de implementación y testing, se creó un repositorio

Descripción	Estado	Prioridad
Agregar la posibilidad de definir un material por defecto al dibujar barras	Resuelta	1
Agregar sprites y flechas a los nodos cuando se definen fuerzas o condiciones de desplazamiento	Resuelta	1
Agregar funcionalidad para "prender" y "apagar" grillas y fuerzas	Resuelta	2
Agregar funcionalidad Abrir Modelo / Guardar Modelo	En proceso	2
Agregar funcionalidad Nuevo Modelo, que limpia la escena y permite definir las nuevas unidades de medida	Para hacer	3

Figura 2: Planilla Excel utilizando metodología Kanban

en Github con el esqueleto de la aplicación y todo código reusable proveniente de la etapa de prototipación. Como metodología de trabajo se utilizó la metodología ágil Kanban. Kanban es un método para gestionar el trabajo intelectual, con énfasis en la entrega justo a tiempo, mientras no se sobrecargan a los miembros del equipo. En este enfoque, el proceso, desde la definición de una tarea hasta su entrega al cliente, se muestra para que los participantes lo vean y los miembros del equipo tomen el trabajo de una cola.

Existen diversas herramientas on-line de planificación y gestión de proyectos, tales como Jira o TFS. Sin embargo, debido a la poca cantidad de personas involucradas en el proyecto (2 desarrolladores y 2 clientes) y a que las tareas a realizar estaban bien definidas, se optó por utilizar una herramienta simple y natural: una planilla Excel online. La misma se encontró en todo momento de libre acceso y modificación para los 4 participantes, y cada tarea tenía asignada una descripción, un estado, y una prioridad.

En un principio, se agregaron todas las tareas a realizar, y ambos desarrolladores tomaban cada una de ellas marcando su estado como «En proceso». Una vez finalizada, se marcaba la tarea como terminada y se subían los cambios al repositorio, marcando cada subida con la funcionalidad correspondiente.

A su vez, los clientes, los cuáles también tenían acceso a la última versión del IETFEM, iban relevando en la misma planilla problemas o cosas a mejorar que se encontraban en el producto, los cuáles pasaban a ser parte de nuestra «pizarra de kanban» y seguían el mismo flujo que las demás tareas.

3.3. Estimación y esfuerzo efectivo

La planificación del tiempo se realizó tomando en cuenta el desconocimiento inicial del problema de cálculo de estructuras y la dificultad de la programación gráfica en 3D. En la Figura 3 se pueden ver las estimaciones realizadas calculando 15 horas de trabajo semanal por desarrollador. Se puede apreciar que el período de trabajo se calculó entre Abril y Diciembre, logrando un total de 34 semanas de trabajo, que se traducen en un total de 1020 horas de trabajo.

Podemos ver también que ciertas etapas se planificaron en simultáneo por ciertos períodos de tiempo, especialmente en las etapas tempranas del proyecto

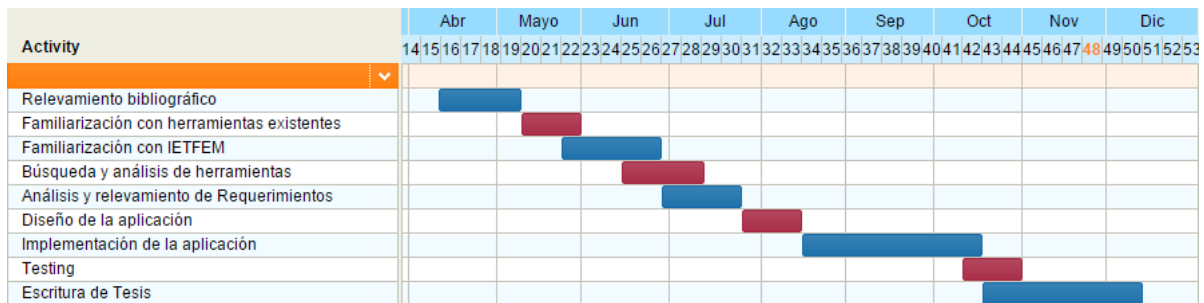


Figura 3: Diagrama de Gantt con la planificación del proyecto

366 donde se comenzó utilizando y comprendiendo tanto el IETFEM como otras
367 herramientas, mientras se iba definiendo al mismo tiempo cómo realizar la in-
368 terfaz. Se planificó de esta manera debido a que se consideró que sería bueno
369 evaluar varias herramientas en simultáneo, a modo de comparar y definir qué
370 funcionalidades y características nos gustaría que estén presentes en nuestro sis-
371 tema. También evaluar cómo llevarlas a cabo utilizando las herramientas que
372 existen en el mercado y el contexto académico en el cuál se quiere insertar la
373 aplicacación.

374 También se observa la concurrencia de tareas en los instantes finales del desa-
375 rrollo, donde se planificó al mismo tiempo el testing y la escritura de la Tesis.
376 Debido a la metodología ágil elegida y al tiempo estipulado, resulta conveniente
377 que el testeo de la aplicación comience cuanto antes, ya que corregir un error
378 pasará a ser parte de nuestra cola de tareas, y dependiendo de la prioridad de
379 la misma podría ser resuelta antes que otras tareas definidas anteriormente pe-
380 ro con una baja prioridad. La escritura de la tesis se planificó en simultáneo
381 simplemente para intentar reducir el tiempo total del proyecto.

382 El cronograma estimado se realizó de manera exitosa, siguiendo cada etapa en el
383 orden estipulado sin demoras excesivas. Como agregado, durante la implemen-
384 tación se descubrieron nuevas funcionalidades que serían útiles en el sistema, las
385 cuales fueron evaluadas con los clientes y algunas de ellas se llevaron a cabo sin
386 problemas, debido a que la metodología de trabajo lo permitía.

387 También es necesario destacar el tiempo invertido en la Ingeniería de Muestra
388 a fines del mes de Octubre, el cuál contempló el diseño de carteles, presentación
389 del proyecto y la propia presencia en el evento. Esto redujo unos días el tiempo
390 estipulado para la escritura de la tesis, el cuál se intenta recuperar durante
391 el mes de noviembre aumentando la cantidad de horas a un promedio de 20
392 semanales por desrrollador dedicadas a dicha tarea.

393 4. Presentación de la solución

394 En esta sección se describe la solución propuesta para el problema planteado,
395 describiendo cada aspecto de la misma y cómo fue realizada cada una de sus
396 funcionalidades. Se detallan además las decisiones que fueron tomadas durante
397 el proceso de análisis y diseño de la aplicación.

398 4.1. Análisis y relevamiento de requerimientos

399 Desde un principio se supo que IETFEM era una herramienta robusta, ofre-
400 ciendo una solución para diferentes problemas posibles. En este sentido, el rele-
401 vamiento de requerimientos se convirtió en una tarea delicada en dónde debía
402 definirse un número acotado de funcionalidades, para un número acotado de la
403 totalidad de problemas que IETFEM podía resolver.

404 Luego de concretar varias reuniones con los clientes, se decidió que la interfaz
405 pueda resolver problemas de estructuras reticuladas, es decir, estructuras for-
406 madas por una serie de vigas entrecruzadas y conectadas entre sí por medio de
407 nudos rígidos. Esto implica que para dibujar una estructura desde la interfaz,
408 el usuario sólo tenga que colocar nodos y barras.

409 El estudiante coloca los nodos en el espacio 3D, y luego define barras entre 2
410 nodos ya dibujados, asignando para cada barra un material que la conforma y
411 el área de su corte transversal, al que llamaremos sección, ambos previamen-
412 te definidos. También pueden definirse ciertas propiedades para cada nodo, en
413 particular, pueden definirse fuerzas aplicadas al mismo, condiciones de despla-
414 zamiento y resortes.

415 Una vez finalizado el proceso de dibujado, se extrae la estructura en un formato
416 reconocible por el motor. Luego se ejecuta el mismo, y se analizan los resultados
417 obtenidos.

418 Destacamos además como funcionalidades secundarias la posibilidad de definir
419 grillas auxiliares con motivo de facilitar el ingreso de datos y la posibilidad de
420 ocultar elementos adicionales, como por ejemplo, los vectores indicadores de
421 fuerzas aplicadas.

422 Basándonos en esta realidad, se definieron los siguientes casos de uso:

- 423 ■ **Alta, Baja y Modificación de Materiales:** Los materiales se definen
424 en base a 5 propiedades: Nombre, Modulo de Young, Gamma, Alpha y
425 Nu.
- 426 ■ **Alta, Baja y Modificación de Secciones:** La sección es el corte trans-
427 versal de una barra, y para este caso solo interesa conocer su área.
- 428 ■ **Alta, Baja y Modificación de Nodos:** Cada nodo tiene asignado un
429 conjunto de coordenadas espaciales (x,y,z). Además es posible asignar al

- 430 mismo una fuerza aplicada, así también como condiciones de despla-
431 zamiento y resortes en cada coordenada.
- 432 ■ **Alta, Baja y Modificación de Barras:** Cada barra tiene asignado un
433 nodo inicial, un nodo final, un material y una sección.
 - 434 ■ **Alta y Baja de Grillas:** Son «cuadrículas» auxiliares que facilitan el
435 proceso de dibujado. Para cada coordenada se define la cantidad de líneas
436 auxiliares y la separación entre ellas.
 - 437 ■ **Modificar Visualización de Propiedad:** Los nodos con propiedades
438 definidas, como por ejemplo fuerzas aplicadas o resortes, son marcados
439 en la pantalla con vectores o geometrías básicas para ser diferenciados
440 del resto. Esta funcionalidad permite ocultar, mostrar y escalar dichos
441 elementos a gusto del usuario.
 - 442 ■ **Nueva Estructura:** Permite limpiar la pantalla para comenzar una nueva
443 estructura.
 - 444 ■ **Abrir y Guardar Estructura:** Se busca la posibilidad de obtener un
445 archivo con la estructura dibujada, de manera de poder seguir con el tra-
446 bajo realizado en otro momento. También es deseable la carga de dicho
447 archivo en la interfaz, obteniendo la misma estructura en la que se estaba
448 trabajando al momento de guardar.
 - 449 ■ **Extraer Estructura:** A partir del dibujo realizado, se extrae un archivo
450 reconocible por el motor de cálculo con la especificación de la estructura.
 - 451 ■ **Procesar Resultado:** Se trata de procesar el archivo resultante del mo-
452 tor, y actualizar la pantalla con la estructura deformada.
 - 453 ■ **Escalar Deformada:** Debido a que en algunos casos las deformaciones
454 pueden ser tan pequeñas que pueden parecer imperceptibles a la vista, se
455 incluye este caso de uso con el fin de «exagerar» la deformación y poder
456 apreciar mejor los resultados obtenidos.
 - 457 ■ **Colorear Estructura:** Al igual que la funcionalidad anterior, este caso
458 de uso aplica a los resultados obtenidos del motor. Se trata de colorear la
459 estructura en base a los datos obtenidos (por ejemplo, pintar de un color las
460 barras que se comprimen y de otro las que se estiran). También se busca
461 transparentar la estructura original o la deformada, para poder apreciar
462 mejor los cambios entre una y otra.

463 En el anexo ?? se incluye la especificación de cada caso de uso descripto.

464 Puede apreciarse el Modelo de Dominio definido en la Figura 4. Como observa-
465 ciones, se destaca la presencia de la entidad «Deformada», la cuál puede existir
466 o no de acuerdo a si ya se procesaron los resultados obtenidos del motor o si
467 se encuentra en el proceso de dibujado. De esta acotación se desprende el «por
468 qué» de la relación 0..1 - 1 entre las entidades «Deformada» y «Estructura»:
469 mientras se dibuja la estructura todavía no se tiene una deformada definida.

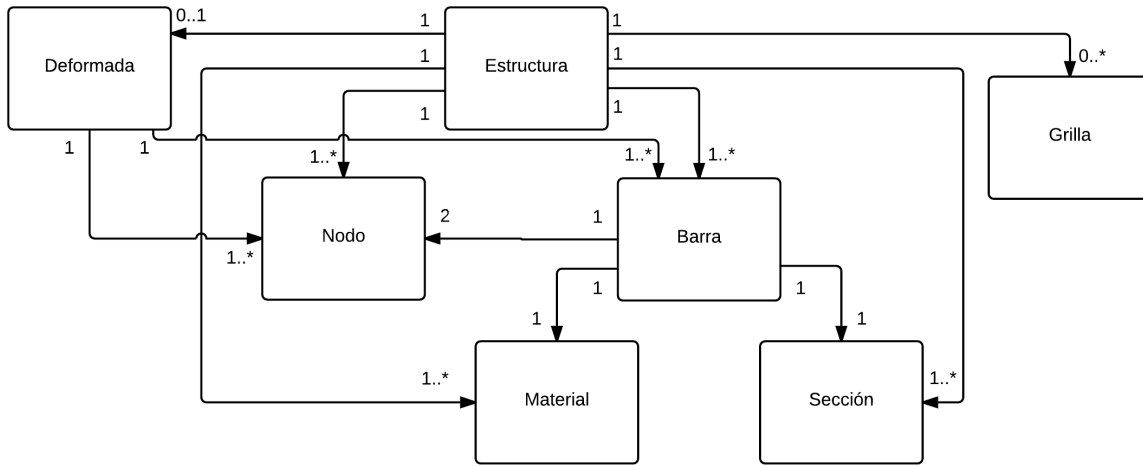


Figura 4: Modelo de dominio de IETFEM

El resto del modelo se encuentra considerablemente intuitivo y adecuado a la realidad planteada.

4.2. Diseño de la solución

4.2.1. Decisiones tomadas

Finalizado el relevamiento de requerimientos y correspondiente análisis, se prosiguió con la etapa de diseño, donde se tomaron decisiones importantes tanto a nivel de diseño tecnológico como en la estructura propia de la aplicación.

Desde las primeras reuniones que se tuvieron con los clientes, el objetivo principal fué lograr una aplicación académica. De esta manera, se tuvo como prioridad mantener la simplicidad y la eficiencia de la herramienta por sobre acomplejizar la misma con funcionalidades potentes que serían útiles en un programa profesional. Por ejemplo, se toma en cuenta que en un ámbito académico, el estudiante no ingresará en el sistema estructuras gigantescas (véase la sección 2.3 del capítulo 5), y sólo utilizará el mismo para los temas comprendidos dentro del curso que desarrolla. Cabe destacar que además, se busca en un segundo plano, lograr el mayor porcentaje de reusabilidad de código posible, ya que en un futuro, IETFEM puede crecer gradualmente para convertirse en un sistema profesional.

Teniendo en cuenta estos aspectos, sumado a las prestaciones destacadas en las herramientas de desarrollo 3D en la web, y la poca cantidad de sistemas de cálculo de estructuras en la nube, se decidió en conjunto con los clientes, realizar

491 la interfaz en un ambiente web.

492 Sin embargo, realizar la interfaz en la nube implica ciertas situaciones preocu-
493 pantes por parte de los clientes, por ejemplo, mantener un servidor donde se
494 aloje la misma una vez finalizado el proyecto. Es necesario entonces destacar
495 ciertas consideraciones sobre la solución elegida:

- 496 ■ Los clientes se sienten a gusto con considerar una versión final en la nube,
497 ya que la mayoría de este tipo de sistemas son de escritorio y no para
498 todos los sistemas operativos.
- 499 ■ Existe preocupación con respecto al servidor donde se aloje la aplicación.
500 En particular, preocupa justamente encontrar un servidor gratuito donde
501 alojarse y cómo mantener la aplicación una vez finalizado el proyecto
- 502 ■ Una de las prestaciones actuales del motor de cálculo existente es que al
503 estar desarrollado en Octave, permite al estudiante ver el funcionamiento
504 interno del código(o incluso programar nuevos cálculos de acuerdo a sus
505 necesidades), enriqueciendo el proceso de aprendizaje de los estudiantes.
506 Dichas características se quieren mantener en la nueva solución.
- 507 ■ Se busca reusabilidad en el código de la aplicación, ya que en un futuro
508 se pretende evolucionar la herramienta a un nivel profesional, donde se
509 pretende que la misma posea diferentes características (por ejemplo, no
510 sería deseable en un sistema profesional que se pueda acceder al código
511 del motor de cálculo directamente).

512 Tomando en cuenta las mencionadas premisas, la solución propuesta es la si-
513 guiente: Se desarrollará la interfaz como una herramienta web. Sin embargo, no
514 se desplegará la misma en un servidor, sino que se encapsulará la misma en un
515 framework que permita ejecutar la misma como una aplicación de escritorio. A
516 ojos del estudiante, la aplicación parecerá ser de escritorio.

517 Una vez finalizado el dibujado de la estructura, el estudiante podrá generar un
518 archivo con al especificación de la estructura, el cual podrá ingresar en el motor
519 de cálculo de manera manual. Luego, puede desde la interfaz procesar la salida
520 del motor para observar sus resultados.

521 De esta manera logramos las siguientes características:

- 522 ■ Para la versión inicial, es decir, la versión académica, se ahorra la utili-
523 zación del servidor, ya que cada sistema ejecutará en la máquina de cada
524 estudiante. Esto implica que el mantenimiento a posteriori sea nulo por
525 parte de los clientes una vez finalizado el proyecto.
- 526 ■ Se desacoplan el motor y la interfaz, o sea, el estudiante puede visualizar
527 los cálculos realizados en el motor, o incluso programar nuevos, sin nece-
528 sidad de tocar el código de la interfaz. Es más, el archivo generado por la
529 interfaz será en un formato legible, lo que hace que le estudiante pueda
530 editar el archivo en caso de agregar cálculos nuevos.

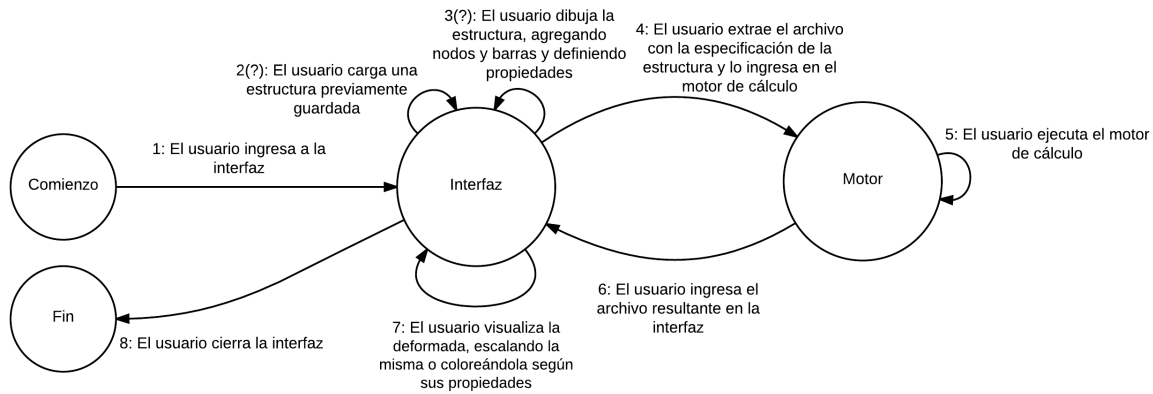


Figura 5: Flujo principal de la aplicación

- 531 ■ A su vez, esta solución no solo permite agregar cálculos nuevos a los estu-
- 532 diantes, sino que permite que el desarrollo del motor siga avanzando sin
- 533 entorpecer la interfaz.
- 534 ■ Se obtiene un código totalmente reusable, ya que si en el futuro se quiere
- 535 evolucionar la herramienta como un producto profesional en la nube, sólo
- 536 basta con desplegar el código de la aplicación en un servidor.

537 De esta manera, la herramienta cumple con todas las especificaciones deseadas
 538 por los clientes, manteniendo las características positivas de la misma, y a su
 539 vez, potenciando la misma en vista de conseguir los objetivos planteados sobre
 540 mejorar la eficiencia y la usabilidad.

541 4.2.2. Diseño final

542 Finalmente, de acuerdo a las pautas establecidas en la sub-sección anterior, el
 543 flujo de la aplicación queda establecido como se muestra en la Figura 5. Los
 544 pasos 2 y 3 se anotan como opcionales debido a que el usuario puede cargar una
 545 estructura guardada como dibujar una nueva. Incluso podría realizar ambas,
 546 editando una estructura guardada antes de ejecutar el motor

547 mostrar un diagrama de diseño

548 hablar de cada modulo

549 4.3. Arquitectura

550 La arquitectura de la aplicación sigue el clásico patrón MVC(Modelo-Vista-
 551 Controlador), donde el usuario se encuentra permanentemente interactuando

552 con el sistema, modificando el Modelo(en este caso, la estructura) y visualizando
553 el mismo en el espacio 3D, al que llamaremos Escena.

554 Cómo se puede ver en la Figura ??, se agruparon los casos de uso relaciona-
555 dos con el fin de crear diferentes subsistemas encargados de realizar cierto tipo
556 de funcionalidades. Cada una de estas componentes, ofrece al usuario diferetes
557 operaciones que afectan tanto el modelo de la estructura que se mantiene al-
558 macenado en la aplicación como lo que se está viendo en el espacio 3D. Por tal
559 motivo, se crearon las componentes «Modelo» y «Escena», las cuáles uniformi-
560 zan todas las operaciones básicas que se hacen en el modelo de la estructura, y
561 en el dibujo en la escena, respectivamente.

562 mostrar la arquitectura mediante diagramas de distribucion de componentes y
563 físicas

564 mostrar ademas como quedaria en version servidor

565 distribucion de componentes: hablar de cada componente: manejo del espacio,
566 del modelo, edicion de puntos, de lineas etc

567 4.4. Tecnologías y herramientas utilizadas

568 4.4.1. HTML5 - Javascript - CSS3

569 dedicar un parrafo a cada una y como lo usamos particularmente en nuestro
570 proyecto

571 4.4.2. Bootstrap

572 que es y como lo utilizamos en nuestro proyecto

573 que mejoras se tienen con respecto a no utilizarlo

574 4.4.3. AngularJS

575 lo mismo que arriba

576 4.4.4. ThreeJS

577 mencionar que es la libreria principal

578 como funciona, habar que etsa sobre ebg1, que funcione en un canvas, que se
579 programa mediante javascript, etc

580 enque lo usamos en el proyecto

581 mencionar y mostrar otros proyectos con three

582 **4.4.5. Electron**

583 mencionr que nosparecio esencial para la prolijidad del proyecto "local", ya que
584 si aspiramos a que el estudiante que lo usa no tenga que lidiar con cosas de
585 cimpuacion, seria contraproducente que tenga que abrir un html pelado en el
586 navegador, donde puede tener problemas de compatibilidad con navegadores
587 diferentes, etc

588 **4.5. Manejo del espacio 3D**

589 **4.5.1. Eventos de usuario**

590 como definimoslo que puede hacer el usuairo
591 los eventos definidos para el mouse y latecla de escape
592 que componentes se comunican directamente con las acciones de usuario y des-
593 cribir .el camino"que realia cada una de ellas

594 **4.5.2. Adición, sustracción y transformación de objetos**

595 hablar de la escena, como se agregan o modifican objetos
596 como se conforma un objeto, atributos relevates
597 hablar del renderiado
598 acciones que se pueden hacer desde el programa

599 **4.5.3. Manejo de la cámara**

600 hablar de orbit controls
601 hablar de elcambio de ejes de coordenadas para que el queda arriba
602 hablar como se mueve la camara tratar de ver el js asqueroso y deducir mate-
603 maticamente como se mueve con respecto al espacio 3d para dibujarlo o mostrar
604 un diagramita

605 **4.5.4. Trazado de rayos e intersecciones con objetos**

606 hablar de como resolvimos la interseccion del 'click'con el rayo de la camara para
607 seleccioar objetos
608 hacer formulacion matematica
609 hablar de las cosas que three automatia y de como qedo resuelto

610 que compones se enargan de esto

611 **4.5.5. Performance**

612 hablar de los probelmas que encontramos al probar la torre eiffel

613 describir que se investigo y se hicieron camnios

614 hablar de los geometry de los objetos

615 de sacar a seleccion al hacer hover

616 de otras performances que se hicieron

617 hablar finalmente de otras mejoras que sepodrian hacer pero resultarian innece-
618 sarios porque serian muy complejos par este proyecto en el que los estudiantes
619 nunca vana hacer una estructura tan grande

620 **4.6. Manejo de datos**

621 hablar de como es la estructura que guardamos

622 **4.6.1. Entrada de información (dibujado e importación)**

623 que informacino guardamos de cada elemento y porque

624 en que momento agregamos cosas al mdoelo

625 hablar del dibujado

626 hablar de la importacion(como reemplamos el modelo)

627 hablar de abrir y guardar proyectos(como reemplamos el modelo)

628 **4.6.2. Mantenimiento de la estructura durante el proceso de dibu-** 629 **jado**

630 aca hablamos de como manipulamos los objetos, modificar y eliminae

631 hablar de mantener la consistencia, al abrir un modelo nuevo, guardar, abrir,
632 etc...

633 **4.6.3. Almacenamiento de la estructura**

634 hablar de que el modelo se va guardando en una variable javascript

635 hacer un estudio sobre que tan eficiente es y si la memoria del navegador
636 'da' para almacenar algo así
637 mostrar un mini ejemplo y exactamente qué se guarda

638 **4.6.4. Salida de Datos**

639 hablar de lo que se genera desde la ui
640 como se genera, proceso que hace el usuario para generarlo
641 validaciones que se toman en cuenta
642 como manipulamos el modelo para generar el txt

643 **4.7. Análisis de resultados del Core**

644 **4.7.1. Generación de resultados**

645 hablar sobre qué genera el core
646 mencionar las cosas que agrega el texto

647 **4.7.2. Introducción de datos en la UI**

648 como se ingresan
649 como se modelan y almacenamos los datos - deformedmodel

650 **4.7.3. Visualización**

651 qué se ve
652 hablar de las opciones que se tienen
653 como hicimos el switch entre deformada en indeformada
654 como hicimos el escalamiento
655 como hicimos el colorado

656 **5. Resultados obtenidos**

657 **5.1. Comparación IETFEM con y sin UI**

658 **5.1.1. Análisis del impacto en la usabilidad**

659 hablar de opinion de estudiantes, posiblemente en la idm

660 mostrar unto por punto en que aspectos se mejoraron

661 **5.1.2. Análisis del impacto en el tiempo de ejecución**

662 usar ietfem viejo y nuevo y calcular tiempos

663 ver si es muy dificil hacer un mini servidor para hacer una comparaciond de
664 tiempos mejor

665 **5.2. Casos de prueba**

666 **5.2.1. Estudio de casos de pequeño porte (Torre pequeña)**

667 hablar de que comenamos con ese

668 se utilio para realiar la primera integracion con el core

669 sedescurbrieron probelmasde ejes y se resolvieron facilmente

670 **5.2.2. Estudio de casos de mediano porte (Grúa)**

671 se intento realiar el caso inicialmente como una prueba de stress

672 no se encontraron problemas

673 cuando se constato que funcionaba bien, se decidio utiliar un caso mas grande

674 utiliado en la idm para mostrar le funcionamiento

675 hablar de que ya se considera exitoso que funcione bien para la grua ya que lso
676 estudiantes nunca van a hacer algo tan grande

677 **5.2.3. Estudio de casos de gran porte y pruebas de stress (Torre 678 Eiffel)**

679 comentar que se decidio hacer latorre eiffel para ver como respondia el programa

680 hablar del trabajo de la perfomrmance y memoria

681 cuanto se mejoro luego de los arreglos
682 importancia de que ande 'perfecto'ya que es un caso inalcanable

683 **6. Conclusiones y trabajo futuro**

684 **6.1. Conclusiones**

685 hablar si las estimaciones y el esfuerxo fueron acertados
686 si se cumplieron los objetivos
687 evaluar la herramienta

688 **6.2. Trabajo a futuro**

689 **6.2.1. Trabajo en el motor**

690 que se puede agregar en el motor
691 porticos
692 osibilidad de migrar a otro lenguaje e integrar en un solo proyecto con la ui

693 **6.2.2. Trabajo en la interfaz**

694 agregar cosas que ya se pueden hacer enel core
695 mejoras de performance
696 otros 'chiches'que tienen programar comerciales
697 delegar responsabilidades a aotra aplicacion

698 **6.2.3. Despliegue de la aplicación**

699 hablar del servidor
700 como funcionaria con servidor y porque no se hio asi
701 donde se podria alojar
702 mejoras que implicari en el sistema

703 **7. Anexos**

704 mini Manual de uso

705 ejemplos de estructuras

706 modelo de dominio

707 casos de uso

708 diagramas de flujo

709 diagramas de arquitectura

710 mas info threajs

711 masinfo otros proyectos similares

712 otras cosas XD