

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

Rafael Olivera - Federico García

27 de noviembre de 2015



# Índice

<b>1. Introducción</b>	<b>6</b>
1.1. Definición del problema y motivación . . . . .	6
1.2. Desarrollo previo . . . . .	6
1.3. Objetivos y resultados esperados . . . . .	8
1.4. Desarrollo del proyecto . . . . .	8
1.5. Organización del documento . . . . .	9
<b>2. Estado del Arte</b>	<b>10</b>
2.1. Cálculo de estructuras . . . . .	10
2.1.1. Cálculo implicados . . . . .	10
2.1.2. IETFEM . . . . .	10
2.1.3. Herramientas comerciales . . . . .	10
2.2. Desarrollo 3D . . . . .	10
2.2.1. OpenGL . . . . .	10
2.2.2. Java 3D . . . . .	10
2.2.3. WebGL . . . . .	10
2.2.4. Otras herramientas . . . . .	10
2.3. Desarrollo 3D en la Web . . . . .	11
2.3.1. HTML5 - Canvas . . . . .	11
2.3.2. Librerías para desarrollo 3D . . . . .	11
2.3.3. Interacción con el usuario . . . . .	11
2.4. Información complementaria . . . . .	11
2.4.1. Investigación sobre proyectos similares en América Latina . . . . .	11
2.4.2. Herramientas de cálculo de estructuras en la Web . . . . .	11
<b>3. Organización del trabajo</b>	<b>11</b>
3.1. Alcance . . . . .	11
3.2. Metodología de trabajo . . . . .	13
3.3. Estimación y esfuerzo efectivo . . . . .	13
<b>4. Presentación de la solución</b>	<b>14</b>
4.1. Análisis y relevamiento de requerimientos . . . . .	14
4.2. Diseño de la solución . . . . .	14
4.2.1. Decisiones tomadas . . . . .	14
4.2.2. Diseño final . . . . .	14
4.3. Arquitectura . . . . .	14
4.4. Tecnologías y herramientas utilizadas . . . . .	14
4.4.1. HTML5 - Javascript - CSS3 . . . . .	14
4.4.2. Bootstrap . . . . .	14
4.4.3. AngularJS . . . . .	15
4.4.4. ThreeJS . . . . .	15

4.4.5.	Electron . . . . .	15
4.5.	Manejo del espacio 3D . . . . .	15
4.5.1.	Eventos de usuario . . . . .	15
4.5.2.	Adición, sustracción y transformación de objetos . . . . .	15
4.5.3.	Manejo de la cámara . . . . .	15
4.5.4.	Trazado de rayos e intersecciones con objetos . . . . .	16
4.5.5.	Performance . . . . .	16
4.6.	Manejo de datos . . . . .	16
4.6.1.	Entrada de información (dibujado e importación) . . . . .	16
4.6.2.	Mantenimiento de la estructura durante el proceso de dibujado . . . . .	16
4.6.3.	Almacenamiento de la estructura . . . . .	16
4.6.4.	Salida de Datos . . . . .	17
4.7.	Análisis de resultados del Core . . . . .	17
4.7.1.	Generación de resultados . . . . .	17
4.7.2.	Introducción de datos en la UI . . . . .	17
4.7.3.	Visualización . . . . .	17
<b>5.</b>	<b>Resultados obtenidos</b>	<b>17</b>
5.1.	Comparación IETFEM con y sin UI . . . . .	17
5.1.1.	Análisis del impacto en la usabilidad . . . . .	17
5.1.2.	Análisis del impacto en el tiempo de ejecución . . . . .	17
5.2.	Casos de prueba . . . . .	17
5.2.1.	Estudio de casos de pequeño porte (Torre pequeña) . . . . .	17
5.2.2.	Estudio de casos de mediano porte (Grúa) . . . . .	18
5.2.3.	Estudio de casos de gran porte y pruebas de stress (Torre Eiffel) . . . . .	18
<b>6.</b>	<b>Conclusiones y trabajo futuro</b>	<b>18</b>
6.1.	Conclusiones . . . . .	18
6.2.	Trabajo a futuro . . . . .	18
6.2.1.	Trabajo en el motor . . . . .	18
6.2.2.	Trabajo en la interfaz . . . . .	18
6.2.3.	Despliegue de la aplicación . . . . .	18
<b>7.</b>	<b>Anexos</b>	<b>19</b>



# **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 descriptas 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 descripta 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 dibujo 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.

## **1.2. Desarrollo previo**

Como mencionamos anteriormente, la FING cuenta con un motor de cálculo de estructuras desarrollado por los Ingenieros Pablo Castrillo y Jorge Pérez. El

mismo resuelve problemas de cálculo de estructuras utilizando el Método de Elementos Finitos(MEF).

El MEF es, desde mediados del siglo XX, una de las principales herramientas utilizadas por los ingenieros para el análisis de sistemas estructurales, mecánicos, eléctricos, etc. El avance de la computación y la disponibilidad creciente de computadores potentes a bajo costo ha provocado que los programas comerciales de MEF para el cálculo estructural sean utilizados masivamente. De hecho, en los últimos cuarenta años el MEF ha transformado los procedimientos de trabajo de todas las áreas de ingeniería y constituye hoy una de las herramientas indispensables con las que un ingeniero debe contar en el ejercicio de su profesión. Por otra parte, el uso del MEF por parte de profesionales no debidamente capacitados podría eventualmente producir errores en el diseño de estructuras, y por tanto, riesgos para los usuarios.

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

De esta manera surge entre docentes del Grupo de Mecánica de Sólidos Computacional (MSC) del Instituto de Estructuras y Transporte (IET) la motivación de brindar una solución al problema a través del desarrollo de un software educativo adecuado: IETFEM.

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

Se buscó una herramienta que sin ser compleja para su aplicación en cursos de grado, permita al estudiante visualizar el funcionamiento interno del método de cálculo. Por ello se optó por la sintaxis de programación de GNUOctave (herramienta libre de alta compatibilidad con Matlab), ya conocida por los estudiantes. Se considera que contar con un software abierto donde los estudiantes pueden entender e incluso programar nuevos cálculos de acuerdo a sus necesidades, enriquece el trabajo desde el punto de vista didáctico.

La forma de ingreso de datos se eligió de acuerdo a otros programas de cálculo de estructuras como SAP2000 donde se deben definir: materiales, secciones, estados de carga, geometrías, conectividades, etc. En el IETFEM se optó por

una entrada de archivo de texto plano donde el estudiante debe ingresar esta información. La salida también es en texto plano (.txt y .tex) y gráfica, al igual que en los programas comerciales.

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

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

### **1.3. Objetivos y resultados esperados**

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

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

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

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

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

### **1.4. Desarrollo del proyecto**

El proyecto comenzó con una fase fuerte de investigación. Inicialmente se realizaron reuniones ocasionales con los clientes, donde se reunió información



valiosa sobre el problema de cálculo de estructuras y el método de elementos finitos. Además se definió qué tipo de herramienta se quería, qué funcionalidades eran deseadas y qué objetivos se buscaban. Durante esta etapa se utilizó el motor de cálculo directamente para comprender su funcionamiento y compararlo con otras herramientas comerciales.

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

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

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

## 1.5. Organización del documento

El resto del documento se organiza de la siguiente manera:

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

Posteriormente, en el capítulo 3, hablaremos de la organización del trabajo a lo largo del proyecto. Hablaremos del alcance del mismo, definiendo las funcionalidades y características específicas que se buscan en el producto final. Se describirá la metodología de trabajo utilizada y se realizarán estimaciones para cada tarea comprendida, comparando finalmente con el esfuerzo efectivo.

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

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

Finalmente, en el 6 capítulo, enumera las conclusiones obtenidas durante el proyecto, analizando el cumplimiento de objetivos y proponiendo posible trabajo a futuro a desarrollar sobre IETFEM.

## **2. Estado del Arte**

### **2.1. Cálculo de estructuras**

#### **2.1.1. Cálculo implícitos**

aca hay que hablar el problema, les mandamos un mail para ver donde podemos leer mas

#### **2.1.2. IETFEM**

hablar de como resuelve el ietfem los problemas de arriba  
beneficios del ietfem, resuelve problemas complejos como los comerciales  
carencias del ietfem, mencionar los problemas un poco mas en detalle, mas que nada haciendo referencia a que a pesar de que soluciona problemas complejos eficientemente, asi como esta es muy dificil de usar

#### **2.1.3. Herramientas comerciales**

Hablar de que centramos la investigacion en 2 grandes productos axis y sap 2000

hablar de cada uno, quien lo desarrolla, si se puede en que esta hecho  
hablar del impacto mundial, que problemas resuelve, hablar de las cosas que nos gustaria poner en nuestro programa, cual nos gustaria evitar o mejorar

### **2.2. Desarrollo 3D**

#### **2.2.1. OpenGL**

investigar y hablar de open gl, que es, en que se usa, etc...  
hablar de porque no la elegimos, dificultad de uso, poca experiencia, etc...

#### **2.2.2. Java 3D**

lo mismo que al anterior

#### **2.2.3. WebGL**

lo mismo que el anterior  
hablar ademas de que al descubrir esta opcion se nos despertó la idea de hacerlo en

#### **2.2.4. Otras herramientas**

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

## **2.3. Desarrollo 3D en la Web**

### **2.3.1. HTML5 - Canvas**

que es html5

Hablar de las facilidades que da el html5 para cosas 3d mediante el canvas  
posibles conexiones entre ebgl y canvas

### **2.3.2. Librerías para desarrollo 3D**

hablar de las librerías que vimos sobre ebgl, threejs y las otras que estuvimos  
viendo que no me acuerdo

porque elegimos threejs y que beneficios se tienen

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

### **2.3.3. Interacción con el usuario**

como pretendemos que sea la interaccion con el usuario, basandonos en los  
programas comerciales y lo que ofrece la eb

plantearse cambiar el titulo de esta seccion para abarcar mas contenido

## **2.4. Información complementaria**

### **2.4.1. Investigación sobre proyectos similares en América Latina**

buscar proyectos similares en internet, en america latina y el mundo, y com-  
pararlos

comparar lo encontrado con ietfem, y rematar señalando que es el primer  
proyecto de este tipo en sudamerica

### **2.4.2. Herramientas de cálculo de estructuras en la Web**

Investigar si existen

en casod e que existan, hacer una mini comparacion con lo que seria ietfem  
eb

mencionar que no existan muchas herramientas y que tendria mucho poten-  
cial

## **3. Organización del trabajo**

### **3.1. Alcance**

Como mencionamos anteriormente, los objetivos planteadas en este proyecto  
se basan en mejorar tanto la eficiencia como la usabilidad del IETFEM. En ese  
sentido, existen dentro de la rama del cálculo de estructuras una infinidad de  
funcionalidades y mejoras posibles que pueden resultar útiles en nuestro sistema.

Por lo tanto, se definió un conjunto acotado de funcionalidades y características deseables en el producto final, apuntando a alcanzar satisfactoriamente los objetivos planteados y lograr una herramienta de alto nivel.

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

La principal y más grande funcionalidad que se desarrollará será la presencia de un espacio 3D. El mismo será el elemento central de la aplicación, mediante la cual el usuario efectuará la mayor parte de las interacciones posibles. Se pretenden integrar dentro de este espacio 3D las siguientes funcionalidades:

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

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

- Abrir y Guardar Estructuras.
- Definición de Materiales.
- Definición de Secciones.

- Asignar propiedades a barras: Material y sección.
- Asignar propiedades a nodos: Fuerzas, Apoyos y Resortes.
- Selección de nodos y barras: Para setear propiedades en las mismas.
- Eliminación de nodos y barras.
- Prendido y apagado de elementos auxiliares
- Seteo de Factor de escalamiento para la estructura deformada

Si bien estos elementos nos permiten estimar una interfaz gráfica completa e intuitiva, resta definir aún la funcionalidad más importante del proyecto: la comunicación con el motor de cálculo. La salida de la interfaz debe ser un archivo reconocible por el IETFEM, del cuál pueda obtener todos los datos de 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. Se hará hincapié en cómo se resolvió esta comunicación en el siguiente capítulo.

En la siguiente figura se pueden apreciar las funcionalidades descriptas y el flujo deseado entre el motor y la interfaz.

### 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 mas 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 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

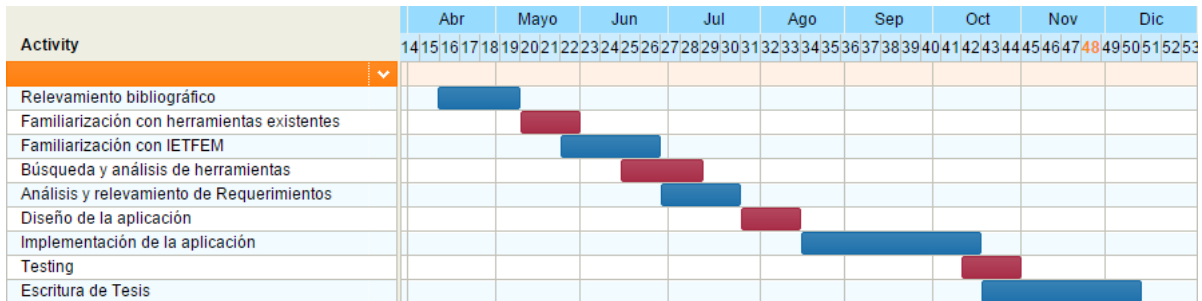


Figura 1: Diagrama de Gantt con la planificación del proyecto

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 Trello. 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 ?? 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 donde se comenzó utilizando y comprendiendo tanto el IETFEM como otras herramientas, mientras se iba definiendo al mismo tiempo cómo realizar la interfaz. Se planificó de esta manera debido a que se consideró que sería bueno

evaluar varias herramientas en simultáneo, a modo de comparar y definir qué funcionalidades y características nos gustaría que estén presentes en nuestro sistema. También evaluar cómo llevarlas a cabo utilizando las herramientas que existen en el mercado y el contexto académico en el cuál se quiere insertar la aplicación.

También se observa la concurrencia de tareas en los instantes finales del desarrollo, donde se planificó al mismo tiempo el testing y la escritura de la Tesis. Debido a la metodología ágil elegida y al tiempo estipulado, resulta conveniente que el testeado de la aplicación comience cuanto antes, ya que corregir un error pasará a ser parte de nuestra cola de tareas, y dependiendo de la prioridad de la misma podría ser resuelta antes que otras tareas definidas anteriormente pero con una baja prioridad. La escritura de la tesis se planificó en simultáneo simplemente para intentar reducir el tiempo total del proyecto.

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

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

## **4. Presentación de la solución**

### **4.1. Análisis y relevamiento de requerimientos**

hablar de las reuniones que tuvimos en la que explicaron el problema y fuimos deduciendo los casos de uso, mencionarlos

hacer una lista de requerimientos

referenciar los casos de uso en el anexo

### **4.2. Diseño de la solución**

#### **4.2.1. Decisiones tomadas**

HABLAR DE PORQUE LO HICIMOS EB Y LOCAL

hablar de las reuniones interminables sobre el servidor y sus preocupaciones por el mantenimiento a futuro y otras complicaciones

hablar brevemente de electron como posible alternativa

hablar de las decisiones de no hacer grandes deformaciones, de hacer solo reticulados de los atributos que no tomamos en cuenta, etc

#### **4.2.2. Diseño final**

- hablar de que dadas las condiciones decidimos hacerlo eb pero manteniendo todo estatico, para que despues no haya problemas de compatibilidad
- mostrar un diagrama de diseño
- hablar de cada modulo

### **4.3. Arquitectura**

- mostrar la arquitectura mediante diagramas de distribuciond e componentes y fisicas
- mostrar ademas como quedaria en version servidor
- distribucion de componentes: hablar de cada componente: manejo del espacio, del modelo, edicion de puntos, de lineas etc

### **4.4. Tecnologías y herramientas utilizadas**

#### **4.4.1. HTML5 - Javascript - CSS3**

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

#### **4.4.2. Bootstrap**

- que es y como lo utilizamos en nuestro proyecto
- que mejoras se tienen con respecto a no utilizarlo

#### **4.4.3. AngularJS**

- lo mismo que arriba

#### **4.4.4. ThreeJS**

- mencionar que es la libreria principal
- como funciona, habar que etsa sobre ebgl, que funcione en un canvas, que se programa mediante javascript, etc
- enque lo usamos en el proyecto
- mencionar y mostrar otros proyectos con three

#### **4.4.5. Electron**

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



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

### **4.5.1. Eventos de usuario**

como definimos lo que puede hacer el usuario  
los eventos definidos para el mouse y teclado de escape  
que componentes se comunican directamente con las acciones de usuario y describir "el camino" que realiza cada una de ellas

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

hablar de la escena, como se agregan o modifican objetos  
como se conforma un objeto, atributos relevantes  
hablar del renderizado  
acciones que se pueden hacer desde el programa

### **4.5.3. Manejo de la cámara**

hablar de orbit controls  
hablar de el cambio de ejes de coordenadas para que el queda arriba  
hablar como se mueve la cámara tratar de ver el js asqueroso y deducir matemáticamente como se mueve con respecto al espacio 3d para dibujarlo o mostrar un diagrama

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

hablar de como resolvimos la intersección del 'click' con el rayo de la cámara para seleccionar objetos  
hacer formulación matemática  
hablar de las cosas que three automatiza y de como quedó resuelto  
que componentes se encargan de esto

### **4.5.5. Performance**

hablar de los problemas que encontramos al probar la torre eiffel  
describir que se investigó y se hicieron cambios  
hablar de los geometry de los objetos  
de sacar a selección al hacer hover  
de otras performances que se hicieron  
hablar finalmente de otras mejoras que se podrían hacer pero resultarían innecesarios porque serían muy complejos para este proyecto en el que los estudiantes nunca van a hacer una estructura tan grande

## **4.6. Manejo de datos**

hablar de como es la estructura que guardamos

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

que informacino guardamos de cada elemento y porque  
en que momento agregamos cosas al mdoelo  
hablar del dibujado  
hablar de la importacion(como reemplaaamos el modelo)  
hablar de abrir y guardar proyectos(como reemplaaamos el modelo)

#### **4.6.2. Mantenimiento de la estructura durante el proceso de dibujado**

aca hablamso de como manipulamos los objetos: que s epueden modificar(agregar propiedades), o eliminar  
hablar de mantener la consistencia, al abrir un modelo nuevo, guardar, abrir, etc...

#### **4.6.3. Almacenamiento de la estructura**

hablar de que el modelo se va guardando en una variable javascript  
hacer un estudio sobre que tan eficiente es y si la memoria del navegador  
'da'para almacenar algo así  
mostrar un mini ejemplo y exactamente qué se guarda

#### **4.6.4. Salida de Datos**

hablar de lo que se genera desde la ui  
como se genera, proceso que ace el usuairo para generarlo  
validaciones que se toman en cuenta  
como manipulamos el modelo ara generar el txt

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

#### **4.7.1. Generación de resultados**

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

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

como se ingresan  
como se modelan y almacenan esos datos - deformedmodel

#### **4.7.3. Visualización**

qué se ve  
hablar de las opciones que se tienen  
como hicimos el sitcheo entre deformada en indeformada  
como hicimos el escalamiento

como hicimos el colorado

## **5. Resultados obtenidos**

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

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

hablar de opinion de estudiantes, posiblemente en la idm  
mostrar unto por punto en que aspectos se mejoraron

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

usar ietfem viejo y nuevo y calcular tiempos  
ver si es muy dificil hacer un mini servidor para hacer una comparaciond de  
tiempos mejor

### **5.2. Casos de prueba**

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

hablar de que comenamos con ese  
se utilio para realiar la primera integracion con el core  
sedescubrieron probelmasde ejes y se resolvieron facilmente

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

se intento realiar el caso inicialmente como una prueba de stress  
no se encontraron problemas  
cuando se constato que funcionaba bien, se decidio utilizar un caso mas grande  
utiliado en la idm para mostrar le funcionamiento  
hablar de que ya se considera exitoso que funcione bien para la grua ya que  
lso estudiantes nunca van a hacer algo tan grande

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

comentar que se decidio hacer latorre eiffel para ver como respondia el pro-  
grama  
hablar del trabajo de la perfomrmance y memoria  
cuanto se mejoro luego de los arreglos  
importancia de que ande 'perfecto'ya que es un caso inalcanable

## **6. Conclusiones y trabajo futuro**

### **6.1. Conclusiones**

hablar si las estimaciones y el esfuerzo fueron acertados  
si se cumplieron los objetivos  
evaluar la herramienta

### **6.2. Trabajo a futuro**

#### **6.2.1. Trabajo en el motor**

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

#### **6.2.2. Trabajo en la interfaz**

agregar cosas que ya se pueden hacer en el core  
mejoras de performance  
otros 'chiches' que tienen programar comerciales  
delegar responsabilidades a otra aplicacion

#### **6.2.3. Despliegue de la aplicación**

hablar del servidor  
como funcionaria con servidor y porque no se hizo asi  
donde se podria alojar  
mejoras que implicari en el sistema

## **7. Anexos**

mini Manual de uso  
ejemplos de estructuras  
modelo de dominio  
casos de uso  
diagramas de flujo  
diagramas de arquitectura  
mas info threejs  
mas info otros proyectos similares  
otras cosas XD