



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

ESTUDIO, DISEÑO Y GENERACIÓN DE UN PROTOTIPO PARA LA EDICIÓN DE PATRONES DE FRACTURA ÓSEAS

Alumno: Gema Parra Cabrera

Tutor: Prof. D. Juan José Jiménez Delgado
Dpto: Informática

Julio, 2019



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Juan José Jiménez Delgado, tutor del Proyecto Fin de Carrera titulado:
“Estudio, diseño y generación de un prototipo para la edición de patrones de
fractura óseas”, que presenta Gema Parra Cabrera, autoriza su presentación
para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Julio de 2019

El alumno:

Los tutores:

Gema Parra Cabrera

Juan José Jiménez Delgado

AGRADECIMIENTOS

La finalización de este Trabajo Fin de Grado pone fin, después de 4 años de trabajo y esfuerzo, al Grado de Ingeniería Informática.

En primer lugar, agradecer al Ministerio de Economía y Competitividad y a la Unión Europea (mediante los fondos FEDER) por dar soporte a mi Trabajo Fin de Grado a través del proyecto de investigación DPI2015-65123-R.

En este proceso he adquirido muchos conocimientos académicos, pero no solo de este índole, sino también otros valores como la amistad, el compañerismo, el trabajo diario, la disciplina académica y el hecho no dejar nunca las tareas para el último día. Todo esto ha sido posible gracias a la colaboración y apoyo de los profesores de la Universidad, aportándome sus conocimientos y su forma de ver la vida, así como compañeros, que se han convertido en más que amigos.

Gracias a mi tutor de este Trabajo de Fin de Grado, Juan José Jiménez Delgado, por orientarme durante estos 4 meses y ayudarme lograr la finalización de este trabajo.

Gracias a mi pareja, por apoyarme y animarme en épocas de exámenes, de entrega de prácticas y de agobios, así como a mi familia, por ayudarme siempre a tranquilizarme antes de los exámenes y por apoyarme siempre aunque las cosas no fueran todo lo bien que podrían.

También agradecer a una persona muy importante en mi vida, mi madre. Sin ella no estaría aquí, acabando la carrera y siendo quien soy. Gracias por ser un pilar fundamental, por intentar siempre hacer que llegue a buen puerto y por enseñarme a creer en mí y a que no hay nada que con esfuerzo y dedicación no se pueda conseguir.

“Nunca sobreestimes el poder de la pasión”, Eve Sawyer

Índice

AGRADECIMIENTOS.....	3
Tabla de ilustraciones.....	6
Índice de tablas.....	8
Índice de ecuaciones.....	8
1. Introducción.....	9
1.1. Resumen.....	9
1.2. Motivación.....	10
1.3. Objetivos.....	11
1.4. Estructura del documento.....	12
2. Antecedentes.....	13
2.1. Fracturas óseas.....	13
2.2. Patrones de fractura.....	15
2.3. Análisis forense.....	19
2.4. Definición de parámetros.....	22
2.5. Herramientas para la edición de patrones de fractura.....	26
3. Diseño del prototipo.....	26
3.1. Metodología de generación del prototipo.....	26
3.2. Entorno de trabajo.....	30
3.2.1. Entorno de trabajo inicial.....	30
3.2.2. Justificación de cambio de entorno de trabajo.....	33
3.2.3. Entorno de trabajo definitivo.....	33
3.3. Ingeniería Software.....	37
3.3.1. Planificación.....	37
3.3.2. Estimación de costes.....	39
3.3.3. Iteraciones.....	42
3.3.3.1. 1ª Iteración. Estudio bibliográfico y documentación.....	42
3.3.3.2. 2ª Iteración. Diseño de la aplicación e interfaz.....	43
3.3.3.3. 3ª Iteración. Diseño de los formatos de exportación.....	46
3.3.3.4. 4ª Iteración. Implementación → Desarrollo de la interfaz.....	47
3.3.3.5. 5ª Iteración. Implementación → Dibujado de líneas.....	51
3.3.3.6. 6ª Iteración. Implementación → Funcionalidad completa del sistema.....	56
3.3.3.7. 7ª Iteración. Implementación → Validador automático.....	61
3.3.3.8. 8ª Iteración. Pruebas de patrones de fractura.....	64
3.3.3.9. 9ª Iteración. Validación de resultados.....	64
3.3.3.10. 10ª Iteración. Generación de patrones de fractura.....	65
3.3.4. Caminos de acción.....	65
3.3.4.1. Creación de un patrón.....	65
3.3.4.2. Ejemplo de generación de un patrón.....	68
3.3.5. Decisiones tomadas en la implementación.....	73
3.3.5.1. Uso de grid.....	73
3.3.5.2. Dibujado de líneas.....	74
3.3.5.3. Dibujado del fragmento.....	75

3.3.5.4.	Restricciones	76
3.3.5.5.	Importación/exportación de patrones.....	78
3.3.5.6.	Pasos de dibujado.....	78
3.3.5.7.	Deshacer/rehacer operaciones	79
3.3.5.8.	Validación.	79
3.4.	Exportación e integración en otros sistemas	80
4.	<i>Validación de los patrones de fractura y usabilidad.....</i>	82
4.1.	Pruebas de usabilidad	82
4.2.	Validación automática	84
4.3.	Validación por expertos	91
4.3.1.	Validación de la herramienta.....	91
4.3.2.	Validación de los patrones de fractura	95
5.	<i>Resultados y discusión.....</i>	99
6.	<i>Conclusiones y aspectos de mejora</i>	106
	<i>Bibliografía</i>	108
	<i>Anexos</i>	109
	ANEXO I. Manual de usuario	109
1.	Introducción.....	110
2.	Pasos para la creación de un patrón de fractura.....	110
2.1.	Paso 1: Dibujado del fragmento o punto de fractura	110
2.2.	Paso 2: Dibujado de las líneas de fractura	112
2.3.	Paso 3: Líneas longitudinales	116
2.4.	Paso 4: Líneas ramificadas de las líneas principales	116
3.	Opciones de la herramienta	117
3.1.	Pestaña “Editar”. Deshacer operaciones	117
3.2.	Pestaña “Editar”. Rehacer operaciones	118
3.3.	Pestaña “Archivo”. Nuevo.....	119
3.4.	Pestaña “Archivo”. Abrir	120
3.5.	Pestaña “Archivo”. Guardar	121
3.6.	Pestaña “Archivo”. Guardar como	121
3.7.	Pestaña “Archivo”. Salir	122
3.8.	Pestaña “Ayuda”	123
3.9.	Pestaña “Ver”	124
3.10.	Pestaña “Validar”	125
	ANEXO II. Cuestionario de usabilidad	127
	ANEXO III. Cuestionario de validación de expertos.....	128
	ANEXO IV. Creación de un ejecutable en xCode e instalación en otros equipos.....	130

Tabla de ilustraciones

Ilustración 1. Fractura.....	11
Ilustración 2. Tipos de fracturas	14
Ilustración 3. Ejemplo de patrón de fractura con ramificaciones	16
Ilustración 4. Líneas longitudinales en un patrón	17
Ilustración 5. Elementos de una fractura.....	18
Ilustración 6. Péndulo de experimentación.....	20
Ilustración 7. Aspecto de impacto en una fractura	23
Ilustración 8. Tipo de líneas de fractura y punto de impacto	24
Ilustración 9. Curvatura de la línea de fractura	24
Ilustración 10. Puntos de ramificación.....	25
Ilustración 11. Metodología Scrum. Fases	27
Ilustración 12. Configuración cMake.....	31
Ilustración 13. Configuración de cMake (II)	31
Ilustración 14. Configuración de las variables de entorno	32
Ilustración 15. Configuración de Visual Studio	32
Ilustración 16. Configuración GTK+	34
Ilustración 17. Hacer el link de las librerías	35
Ilustración 18. Linkear librerías (II).....	35
Ilustración 19. Linkear librerías (III).....	35
Ilustración 20. Diagrama de Gantt	38
Ilustración 21. Storyboard aplicación.....	44
Ilustración 22. Prototipo a papel 1.....	45
Ilustración 23. Prototipo a papel 1.....	45
Ilustración 24. Diagrama de casos de uso de la interfaz.....	48
Ilustración 25. UML diseño de la interfaz y eventos necesarios	50
Ilustración 26. Interfaz del prototipo	51
Ilustración 27. Diagrama de casos de uso del dibujado de la línea	53
Ilustración 28. UML para el dibujado de líneas.....	55
Ilustración 29. Diagrama de clases 6ª Iteración.....	58
Ilustración 30. UML de iteración 6.....	60
Ilustración 31. Diagrama de casos de uso final	62
Ilustración 32. UML final	63
Ilustración 33. Aspecto de la aplicación inicial	68
Ilustración 34. Selección de la forma del fragmento desprendido	68
Ilustración 35. Dibujado del fragmento desprendido.....	69
Ilustración 36. Dibujado de líneas oblicuas	69
Ilustración 37. Dibujado de líneas oblicuas con línea guía.....	70
Ilustración 38. Dibujado de líneas longitudinales	70
Ilustración 39. Paso 4 de dibujado	71
Ilustración 40. Dibujado de líneas secundarias	71
Ilustración 41. Guardado del patrón de fractura	72
Ilustración 42. Patrón de fractura generado importado	72
Ilustración 43. Línea oblicua con línea guía en el primer cuadrante	75
Ilustración 44. Ejemplo de fichero de exportación.....	81
Ilustración 45. Estadísticas de características de una fractura	86
Ilustración 46. Patrón de fractura generados	86
Ilustración 47. Comparativa con respecto al primer test	87

Ilustración 48. Localización de líneas longitudinales según la dirección de impacto	87
Ilustración 49. Fractura lateral con test 2	88
Ilustración 50. Análisis de fractura lateral con test 2	88
Ilustración 51. Test 2 válido	89
Ilustración 52. Media de líneas de fractura según dirección de impacto	89
Ilustración 53. Número de líneas medio por aspecto según dirección de impacto	90
Ilustración 54. Imagen 1 a valoración de expertos	95
Ilustración 55. Imagen 2 sometida a valoración experta	96
Ilustración 56. Imagen 3 sometida a valoración experta	97
Ilustración 57. Imagen 4 sometida a valoración experta	98
Ilustración 58. Imagen 5 sometida a valoración experta	99
Ilustración 59. Imagen 1 a valoración de expertos	100
Ilustración 60. Imagen 2 sometida a valoración experta	101
Ilustración 61. Imagen 3 sometida a valoración experta	102
Ilustración 62. Imagen 4 sometida a valoración experta	103
Ilustración 63. Imagen 5 sometida a valoración experta	104
Ilustración 64. Configuración fragmento. Opciones	111
Ilustración 65. Imagen con punto de fractura	111
Ilustración 66. Patrón con fragmento de viruta	112
Ilustración 67. Configuración de una línea	112
Ilustración 68. Configuración de línea. Tipo	113
Ilustración 69. Configuración de línea. Grosor	113
Ilustración 70. Configuración de línea. Color	113
Ilustración 71. Patrón de fractura	115
Ilustración 72. Antes de deshacer	117
Ilustración 73. Operación deshecha	117
Ilustración 74. Operación deshecha	118
Ilustración 75. Operación rehecha	118
Ilustración 76. Crear un canvas nuevo	119
Ilustración 77. Nuevo canvas creado	119
Ilustración 78. Cargar patrón de fractura	120
Ilustración 79. Patrón de fractura cargado	120
Ilustración 80. Guardado del patrón de fractura	121
Ilustración 81. Guardar como	121
Ilustración 82. Ventana de guardado	122
Ilustración 83. Salir del programa	122
Ilustración 84. Ayuda	123
Ilustración 85. Ventana de ayuda	123
Ilustración 86. Pestaña VER	124
Ilustración 87. Ejemplo pestaña ver lateral	124
Ilustración 88. Pestaña Validar. Características	125
Ilustración 89. Pestaña validar. Líneas longitudinales	125
Ilustración 90. Pestaña validar. Líneas de fractura	126
Ilustración 91. Creación de un esquema	130

Índice de tablas

Tabla 1. Planificación temporal.....	37
Tabla 2. Costes hardware	39
Tabla 3. Costes software	39
Tabla 4. Costes derivados.....	40
Tabla 5. Gastos en recursos humanos	41
Tabla 6. Resultados de test de usabilidad	83
Tabla 7. Resultados test de expertos	93
Tabla 8. Valoración experta VS automática	105

Índice de ecuaciones

Ecuación 1. Distancia de un punto a una recta	77
Ecuación 2. Distancia entre dos puntos.....	77
Ecuación 3. Punto de intersección entre dos líneas	77

1. Introducción

Este capítulo va a ser dedicado a una introducción al documento en el que encontramos un resumen acerca de la temática del proyecto, su motivación y objetivos.

1.1. Resumen

En contadas ocasiones puede realizarse un TAC cuando se produce una fractura ósea para obtener una reconstrucción 2D de los fragmentos y ayudar a los cirujanos en la reducción de la fractura. Con esta información visual de la fractura, el cirujano puede simular el proceso de unir los fragmentos y aplicar una solución, ya sea placas, tornillos o lo que el profesional considere apropiado, para fijar esa fractura y conseguir estabilizarla y que en un futuro se unan los huesos fracturados. Esta ardua tarea requiere grandes dosis de imaginación y entrenamiento. El problema radica en que, a veces, no se encuentran imágenes TAC de fracturas óseas complejas que el cirujano pueda usar para experimentar y entrenarse. El propósito del proyecto es la obtención de diversos tipos de fracturas virtuales, a partir de modelos 3D de huesos sanos, de modo que permitan entrenar a especialistas en la ardua tarea de reducir una fractura.

Uno de los medios para obtener una fractura sería la de generar un patrón de fractura. Este patrón se puede obtener a partir de reconstrucciones 3D de huesos fracturados, o bien, puede diseñarse de modo interactivo por especialistas. El propósito de este TFG sería el de obtener un módulo de diseño de patrones de fractura, mediante la implementación de un editor de fracturas parametrizable, que permitiría diseñar fracturas de distinto tipo por especialistas, que luego podrían aplicarse sobre modelos 3D de huesos, obteniendo a posteriori diversos fragmentos de hueso, simulando una fractura real.

1.2. Motivación

El motivo principal de la realización de este trabajo es la existencia de un grupo de investigación en la Universidad de Jaén, concretamente el TIC-144 Informática Gráfica y Geomática, que está realizando una investigación acerca del diseño y modelado de patrones de fractura. Dicha investigación, comenzó en el año 2016 y tiene prevista su finalización a finales de 2019 y está siendo subvencionada por el Ministerio de Economía y Competitividad y Unión Europea a través de fondos FEDER.

El proyecto de investigación mencionado anteriormente se lleva a cabo debido a que la fracturación de modelos geométricos es un campo de la informática gráfica en auge, en el que existen trabajos importantes pero que se encuentra abierto a investigación, debido sobre todo a la complejidad que representa realizar una simulación física realista en diversos sentidos. Existen aún muchos retos, sobre todo en relación a la validación de los modelos obtenidos tras la fractura, a la incorporación de información física sobre los modelos geométricos, o incluso a su utilización en simulaciones o casos reales, como puede ser en medicina. En este proyecto se propone la investigación básica relacionada con la generación de fracturas sobre modelos geométricos que representan estructuras óseas. Esto supone grandes retos en cuanto a la aplicación de patrones de fractura reales sobre modelos geométricos y su posterior validación tanto desde el punto de vista clínico como desde el punto de vista geométrico. Actualmente no se evidencian en la literatura trabajos que desarrollen estos retos aplicados a modelos óseos y por ello en este Trabajo de Fin de Grado, se va a abordar la generación de patrones de fractura, así como la importación de patrones ya generados para una posterior validación de los patrones de fractura obtenidos, pudiendo así cubrir la falta de sistemas de validación en la actualidad.

1.3. Objetivos

El objetivo de este proyecto es obtener una aplicación de diseño de patrones de fractura mediante la implementación de un editor de fracturas parametrizable, que permita diseñar fracturas de distinto tipo por especialistas.

Un patrón de fractura es una plantilla que permite resumir un tipo determinado de fractura ósea y su aplicación sobre un hueso sano. Tras la aplicación de dicho patrón de fractura se obtendrá una determinada fractura.

Por tanto, este editor permitirá el diseño de patrones de fractura en el plano, consiguiendo un aspecto similar a la figura siguiente.

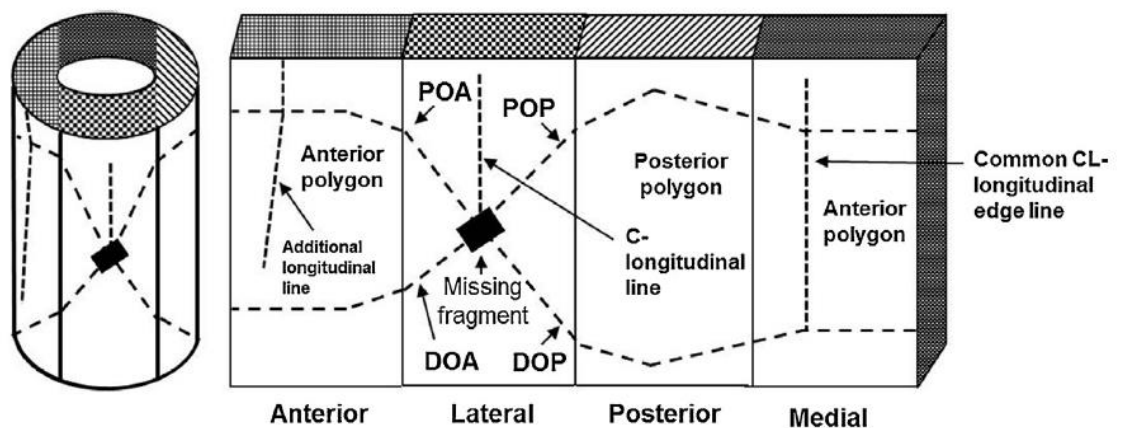


Ilustración 1. Fractura

Los objetivos que este proyecto engloba son los siguientes:

- Estudio, análisis y definición de los tipos de fractura y patrones de fractura.
- Definición de parámetros de diversos tipos de fractura.
- Diseñar un editor de patrones de fractura parametrizable.
- Generación de un prototipo.
- Validación del sistema por expertos.
- Generación de patrones de fractura con el prototipo.

1.4. Estructura del documento

El documento consta de 6 capítulos en los que se organiza todo el contenido de la memoria y anexos.

Capítulo 1:

Se corresponde con la introducción de la memoria, en la que se indica en un breve resumen la temática del proyecto, su motivación y los objetivos que se pretenden conseguir con la realización de la aplicación software.

Capítulo 2:

Recoge toda la información necesaria acerca de las fracturas óseas y de los patrones de fractura para un conocimiento completo sobre dichas fracturas para así lograr la consecución de un producto software acabado y correcto.

Capítulo 3:

En este capítulo se recoge todo lo relativo a la creación del prototipo software. Su contenido va desde la elección de la metodología a seguir, al entorno de trabajo seleccionado, pasando por toda la temática relacionada con la Ingeniería del Software, como es la planificación y las iteraciones por las que pasará el proyecto, así como el proceso de generación de una fractura e integración con otros sistemas.

Capítulo 4:

En este capítulo se describen los procesos de validación de fracturas y de usabilidad de la herramienta, mediante usuarios reales o mediante una validación automática.

Capítulo 5:

Este capítulo describe los resultados obtenidos por la herramienta de validación automática y por los expertos, llegando a realizar un análisis y comparativa de ambos resultados para indicar la validez del sistema.

Capítulo 6:

Corresponde a las conclusiones del proyecto y los aspectos a mejorar para trabajos futuros en la herramienta.

Bibliografía

Contiene la bibliografía consultada en la realización del proyecto.

Anexos

Recoge tres anexos: manual de usuario, test de usabilidad, test de expertos y finalmente exportación del ejecutable a otros sistemas.

2. Antecedentes

Este capítulo se ha basado en el estudio de todo lo relacionado con los patrones de fractura en distintos artículos científicos, obteniendo los conocimientos necesarios en el área para la realización del prototipo.

2.1. Fracturas óseas

Las fracturas es una discontinuidad en los huesos, a consecuencia de golpes, fuerzas o tracciones cuyas intensidades superen la elasticidad del hueso.

En una persona sana, siempre son provocadas por algún tipo de traumatismo, pero existen otras fracturas, denominadas patológicas, que se presentan en personas con alguna enfermedad de base sin que se produzca un traumatismo fuerte. Es el caso de algunas enfermedades orgánicas y del debilitamiento óseo propio de la vejez.

Otra causa de ruptura es el caso en que se aplique más presión sobre un hueso de la que puede soportar, éste se parte o se rompe. Una ruptura de cualquier tamaño se denomina fractura y si el hueso fracturado rompe la piel, se denomina fractura abierta (fractura compuesta).

También puede producirse una fractura por estrés o sobrecarga es una fisura delgada en el hueso que se desarrolla por la aplicación prolongada o repetitiva de fuerza sobre el mismo.

Clasificación de una fractura según su localización:

Los huesos largos se pueden dividir anatómicamente en tres partes principales: la diáfisis, las epífisis y las metáfisis.

- La diáfisis es la parte más extensa del hueso, que corresponde a su zona media. Si una fractura se localiza en esta zona se denomina diafisaria.
- Las epífisis son los dos extremos, más gruesos, en los que se encuentran las superficies articulares del hueso. En ellas se insertan gran cantidad de ligamentos y tendones, que refuerzan la articulación. Si una fractura se produce en esta zona se denomina epifisaria.
- Las metáfisis son unas pequeñas zonas rectangulares comprendidas entre las epífisis y la diáfisis. Sobre ellas se encuentra el cartílago de crecimiento de los niños. Si se produce en esta zona se denomina metafisaria, pudiendo afectar a la zona metáfisis superior o inferior del hueso.

Las fracturas según el trazo de la fractura generalmente se clasifican en 6 tipos clásicos: transverso, oblicuo o mariposa, espiral, segmento y triturado.

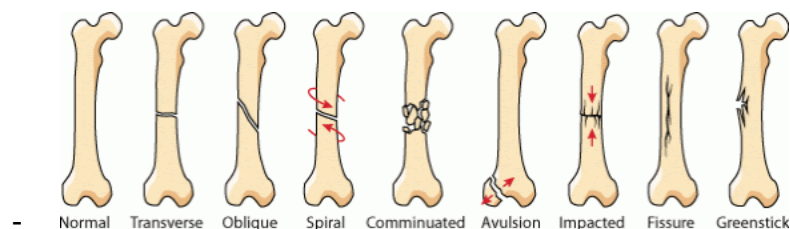


Ilustración 2. Tipos de fracturas

- La fractura **transversal** se extiende en ángulos aproximadamente rectos al eje largo del hueso largo, mientras que las fracturas **oblicuas** se extienden diagonalmente a través de la diáfisis. Generalmente las

fracturas oblicuas tienen líneas de fractura que forman un ángulo mayor o menor de 90 grados con el eje longitudinal del hueso.

- La fractura **longitudinal** se produce con una línea de fractura que sigue el eje longitudinal del hueso.
- En cuanto a las fracturas en **espiral**, sabemos que son causadas por fuerzas de rotación en el hueso o combinación de torsión y flexión, creando comúnmente un corte puro entre transversales paralelos.
- Otro tipo de fractura es la fractura conminuta, que es aquella que genera más de dos fragmentos y múltiples líneas de fractura y, por lo general, es el resultado de niveles de fuerza muy altos.

2.2. Patrones de fractura

En primer lugar, un traumatismo óseo es una fuente importante de información acerca de las circunstancias que produjeron o pudieron producir un determinado accidente. Por ello, si realizamos una interpretación de una fractura, podríamos determinar la ubicación y el lugar de impacto, incluso llegando a poder determinar si el accidente fue una caída, el tipo de superficie de impacto, que tipo de daño sufrió el hueso, etc. El análisis de un patrón de fractura también puede ser una herramienta útil para identificar qué ocurrió en un accidente, ya que existe un número limitado de modos de carga a los que se puede someter un hueso, y esto da lugar a patrones de fractura predecibles. Este es el principal motivo por el que es tan importante el análisis de un patrón de fractura y el tener acceso a una herramienta como la propuesta para generar dichos patrones y poder analizarlos.

Cabe destacar, que cuanto mayor es la magnitud de la fuerza que se le aplica al hueso, mayor es su contenido en energía y, por tanto, mayor es la destrucción ósea que genera. Por el contrario, cuanto más complejo sea el patrón de fractura que encontramos, mayor será la energía necesaria para producir dicha fractura. El análisis de este patrón de fractura puede ser un proceso complicado, por lo que habitualmente se aborda esta tarea mediante experimentación en lugar de con teorías.

Otro aspecto fundamental de estudio en la creación de fracturas ha sido ver que líneas de fractura serían válidas en el dibujado, por lo que he tenido que estudiar mediante casos reales de dichos artículos que características tienen las líneas de fractura de los patrones obteniendo que:

- Las líneas de fractura siempre comienzan a salir a partir del punto de impacto o fragmento de viruta. Si a partir de dicho punto no salen líneas, no puede ser una fractura válida, ya que en el caso de que un impacto produzca una fractura, siempre va a comenzar dicha fractura a partir de donde se ha dado el golpe.
- Una línea de fractura oblicua no se puede entrecruzar con ninguna otra línea de fractura oblicua.
- Por otro lado, se ha estudiado también que en una fractura puede haber líneas oblicuas y que de estar pueden dar la vuelta completa al hueso, volviendo a entrar al fragmento o que simplemente pueden quedarse a la mitad. De igual forma se ha observado que puede haber como mucho dos líneas oblicuas, considerando que una línea oblicua puede pasar dos veces por el fragmento (inicio y fin).
- Un factor importante también en el dibujado de líneas oblicuas es que una línea oblicua siempre sigue la misma dirección, es decir, si va hacia la izquierda, va a seguir siempre hacia la izquierda, en ningún momento va a intentar ir hacia la derecha.
- Se pueden encontrar ramificaciones que salgan de una línea oblicua. Una ramificación de una línea de fractura se puede entrecruzar con cualquier otra ramificación, así como con cualquier otro tipo de línea.

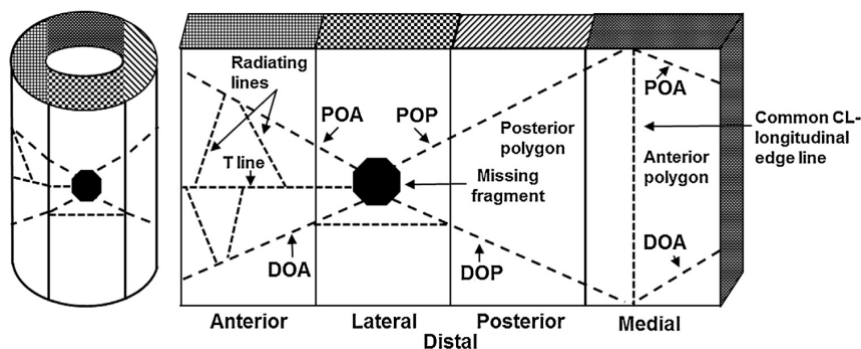


Ilustración 3. Ejemplo de patrón de fractura con ramificaciones

- Las líneas longitudinales si aparecen deben de estar: si es en el aspecto impactado pasando por el fragmento, ya que se originan a partir del fragmento, y si no están en el aspecto impactado las líneas de fractura longitudinales están interconectadas con al menos una línea oblicua debido a que se producen a partir de ella.

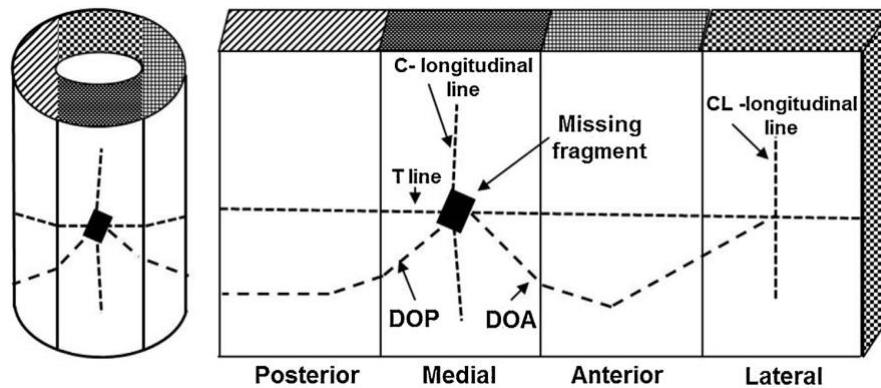


Ilustración 4. Líneas longitudinales en un patrón

Por otro lado, se han revisado y estudiado a fondo las distintas estadísticas dadas como resultado en los diversos test realizados por el equipo implicado en el estudio de patrones de fracturas de los artículos en los cuales me he basado. Obteniendo conclusiones claras acerca de las evidencias dadas por dichas pruebas.

Una de las evidencias claras obtenidas es que a mayor velocidad mayor es el tamaño del fragmento, mayor es la longitud de las líneas de fractura y mayor es el número de líneas e índice de curvatura. En definitiva, la velocidad de impacto es un factor clave en la generación de una fractura mayor o menor.

También se ha podido observar que la velocidad también determina en gran medida la presencia de líneas longitudinales, siendo bastante probable la presencia de líneas longitudinales a baja velocidad, sobre todo en el aspecto impactado y en el contralateral, así como que a una velocidad moderada siempre aparecen líneas longitudinales en la parte contralateral y que a alta velocidad no suelen aparecer en ningún caso este tipo de líneas.

Otros datos también a destacar del estudio es que, dependiendo del aspecto impactado, podemos encontrar distintas estadísticas.

Teniendo en cuenta por ejemplo un análisis general de las características de una fractura, podemos encontrar que para cada tipo de fractura según su dirección de impacto tiene un número de líneas de fractura característico, un tamaño del fragmento de viruta desprendido y una longitud determinada para sus líneas o dicho de otra forma un tamaño distinto para el polígono al que dan lugar sus líneas de fractura.

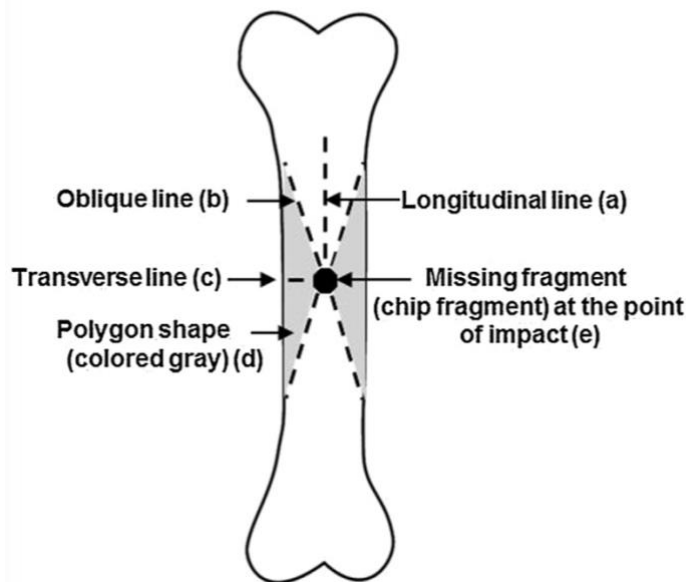


Ilustración 5. Elementos de una fractura

Otro dato importante a tener en cuenta, es que, dependiendo de la dirección de impacto, hay líneas longitudinales que en ningún caso pueden aparecer en nuestra fractura según las estadísticas. Por ejemplo, en impactos laterales no pueden aparecer líneas longitudinales en el aspecto lateral y en el contralateral, en el impacto posterior no pueden aparecer líneas longitudinales en el aspecto contralateral ni en el anterior. En el impacto medial y en el anterior las líneas longitudinales no pueden aparecer en el aspecto anterior.

Un aspecto a destacar de los estudios realizados sobre los test según distintos parámetros es que tienen un valor variable, es decir, trabajan con una

media de todos los casos, pero con una desviación, por lo que para poder contrastar estos datos con los obtenidos por nuestra herramienta tendremos que comparar un rango del 80% para trabajar con un margen y no simplemente con la media, ya que la oscilación de la media con los valores que podrían salir de la desviación es más o menos grande.

2.3. Análisis forense

El artículo “The effect of impact tool geometry and soft material covering on longitudinal fracture patterns in children”[1], junto con otros artículos, es el punto de partida para el estudio de los patrones de fractura que se validan en el editor a crear. Los estudios sobre análisis de fracturas en un contexto forense son de gran importancia en la reconstrucción de la causa y la forma de la muerte en casos como asalto homicida, suicidio, caídas, maltrato infantil y accidentes de tráfico. La interpretación precisa de la fractura es un objetivo y un medio esencial para identificar la ubicación y el número de sitios de impacto, estableciendo la secuencia de golpes y determinando las características del objeto que infligió las lesiones. Hay muchos factores que pueden afectar al patrón de fractura y a su severidad. Dos factores importantes son los tejidos blandos que recubren el hueso y la geometría del elemento impactante. Los tejidos que rodean el hueso, incluyendo los músculos, tendones y ligamentos, entre otros, pueden afectar el patrón de fractura absorbiendo parte de la energía de carga y también creando una carga adicional, que genera que un impacto producido por ejemplo en una pierna cree más fracturas de tipo conminutas que si se tratase de un hueso desnudo. Por otro lado, se han hecho pruebas con huesos de cadáveres de los dos sexos y generalmente el agrietamiento o desgarro del hueso ocurre en el lado convexo y que el aplastamiento en el lado cóncavo del mismo. Sin embargo, la forma del elemento que genera el impacto no se pudo especificar con claridad en dichos estudios.

Hasta ahora solo unos pocos estudios han examinado la asociación entre el tamaño y la geometría del impacto y entre la presencia de tejido blando que recubre el hueso y el patrón de fractura y la severidad de la lesión.

Para estudiar en profundidad las relaciones entre las distintas características que generan una fractura se han hecho numerosos experimentos y se han aplicado distintos métodos.

Uno de los experimentos realizados es el uso de un péndulo de baja energía, llamado Dynatup, Modelo POE 2000 (Instron Co.), que se utilizó para aplicar carga de impacto en los huesos femorales de cerdo. Además, se fabricaron soportes a medida para mantener cada hueso en su lugar durante la carga. El hueso se fijaba en la posición adecuada para producir una flexión de tres puntos con compresión adicional del hueso, simulando así el peso corporal en una pierna. Las fuerzas de compresión se generan al mover las placas que se situaban en los bordes del hueso y se ajustaban y modificaban con el Vishay Tadea-Huntleigh 615 y con el Rinstrum 310, respectivamente.

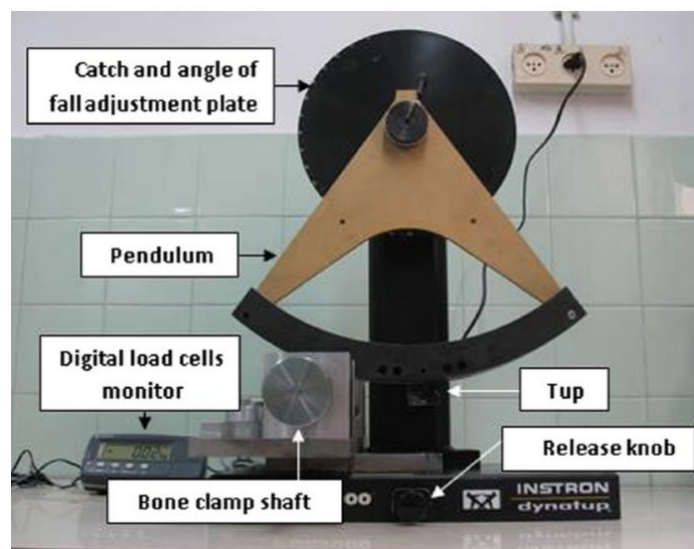


Ilustración 6. Péndulo de experimentación

Para la preparación de los huesos que iban a someter a experimentación, se elegían fémures de cerdos de 5 a 6 meses de edad, debido a la similitud entre los huesos de cerdo y de humanos principalmente en su forma, microestructura y densidad, que permite determinar la masa ósea humana y la fuerza. Después de elegir los huesos, se procede a limpiarlos, quitando todos los tejidos que lo rodean y se examina que el hueso de cerdo no sufra ninguna enfermedad o anomalía y se almacenan a -20° etiquetados. Tras hacer esto, los huesos se

descongelan con agua a temperatura ambiente y se recubren con una solución dejándolos secar y enfriar por 30 minutos.

Una vez están los huesos preparados, se procede a aplicar la carga de flexión en tres puntos con compresión mediante la herramienta mencionada anteriormente. Las pruebas realizadas consistían en usar una tup redondeada sin una interfaz suave, una tup de 10 mm con una interfaz suave, una tup de 10mm sin una interfaz suave, una tup de 20mm con una interfaz suave y una tup de 20mm sin una interfaz suave.

Tras hacer las pruebas, se encontraron diferencias claras entre los huesos después de los impactos con distintos tamaños de objetos que impactaban sobre el hueso. En general, los huesos que se golpean con un cuerpo de impacto grande presentan un patrón de fractura más complicado en un área más grande que los huesos golpeados por un cuerpo más pequeño. El número y la magnitud de las líneas de fractura aumenta con el tamaño del cuerpo de impacto, siendo el aspecto impactado el más afectado. Además, la línea longitudinal que aparece en las fracturas es más larga en los huesos sometidos a impactos con objetos más grandes, teniendo en la mayoría de los casos un tamaño similar al cuerpo impactante. Con el tamaño del cuerpo impactante también aumenta el área de ramificación y el tamaño del fragmento desprendido, aunque si el impacto se produce con un objeto redondo, no se crea la línea longitudinal y las líneas oblicuas emergen desde el punto de impacto.

Por otro lado, la realización de este proyecto podría ser clave para la reproducción y creación de patrones de fractura que más tarde pudieran ser analizados, para así tener otra herramienta que no necesite tanta preparación como la experimentación anteriormente descrita, pero sí permita reproducir patrones de fractura y analizar su validez, pudiendo llegar a esclarecer algunas de las condiciones de fractura.

Además, el hecho de que gracias al editor de fracturas podamos generar patrones de cualquier tipo, así como poder ajustar algunos parámetros para su generación creando un editor parametrizable, hará que se puedan cubrir algunos

de los casos experimentados en los análisis forenses y que se permita una experimentación y análisis de patrones de fractura sin necesidad de hacer procesos tan complicados con huesos reales.

Otro punto a favor de la herramienta es que se pueda validar el patrón generado gracias a la información y parámetros obtenidos de todos los artículos estudiados [1] [2] [3], generando patrones que realmente son correctos. Además, se podrán modificar patrones ya creados anteriormente mediante la importación de los mismos, pudiendo estudiar que cambios en los parámetros y análisis existen si por ejemplo una línea longitudinal ya generada no se hubiera generado en dicho lugar, permitiendo un nuevo foco de análisis de parámetros para determinar las circunstancias de un suceso.

2.4. Definición de parámetros

Tras el análisis de distintos artículos, se han encontrado multitud de parámetros que intervienen en una fractura, pero solo algunos son necesarios, ya que parámetros como la dirección de impacto, la velocidad de impacto, aceleración, fuerza, etc. son parámetros que se determinan a partir del análisis de un patrón, por lo que no son necesarios para la creación del mismo. Los atributos considerados necesarios para la creación de un patrón de fractura y su posterior validación son:

- Dirección del impacto, pudiendo ser recto, oblicuo, lateral, anterior o posterior

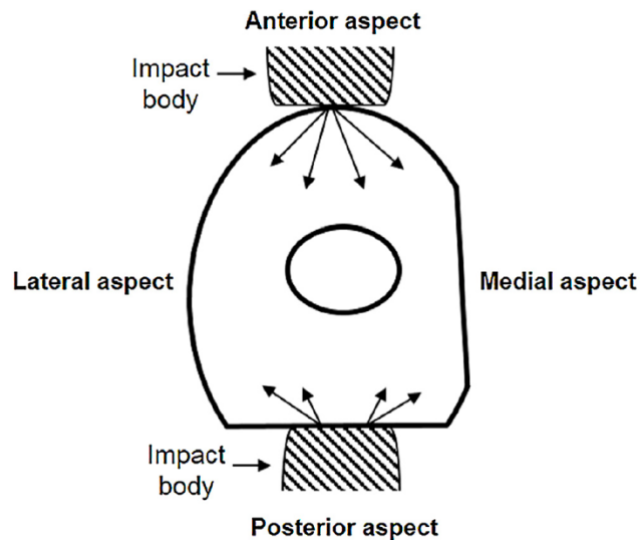


Ilustración 7. Aspecto de impacto en una fractura

- Posición
- Forma del fragmento inicial
- Tipo de línea de fractura:
 - Líneas longitudinales: son líneas rectas que se extienden proximal o distalmente hacia la epífisis. Las líneas longitudinales pueden aparecer en todos los aspectos del hueso, pero generalmente aparecen en el área de impacto, en la cara contralateral o en ambos aspectos.
 - Líneas oblicuas: líneas de fractura que se extienden en el ángulo con respecto al eje largo del hueso. Las líneas oblicuas por lo general se extienden proximal o distalmente hacia la epífisis y hacia otros aspectos del hueso.
 - Poligonal: estas líneas se definen si un área del hueso está rodeada por dos líneas oblicuas. Una línea corre en dirección proximal y otra distal desde el área de impacto, generalmente hacia el aspecto contralateral.
 - Línea transversal: la línea horizontal rodea completamente la diáfisis. Aparece recta o fracturada.

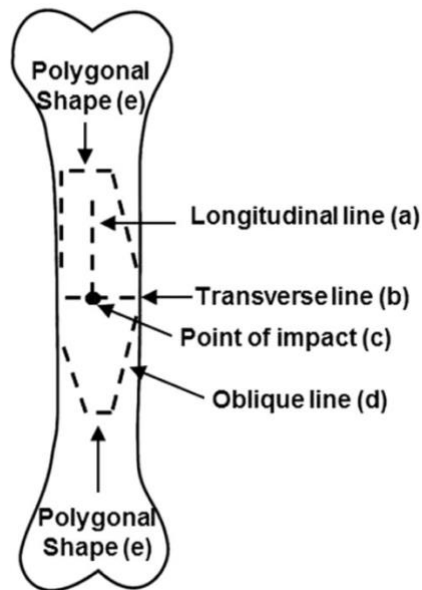


Ilustración 8. Tipo de líneas de fractura y punto de impacto

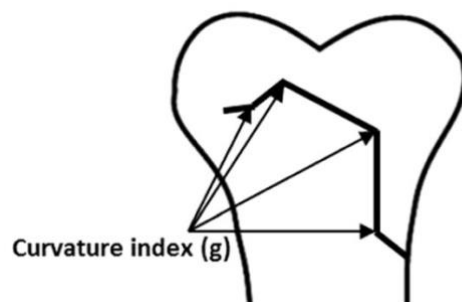


Ilustración 9. Curvatura de la línea de fractura

- Puntos de ramificación: Es un punto a lo largo de una línea longitudinal desde donde se ramificaron dos líneas oblicuas. Las medidas relacionadas con los puntos de ramificación es la longitud entre los puntos de ramificación principalmente.

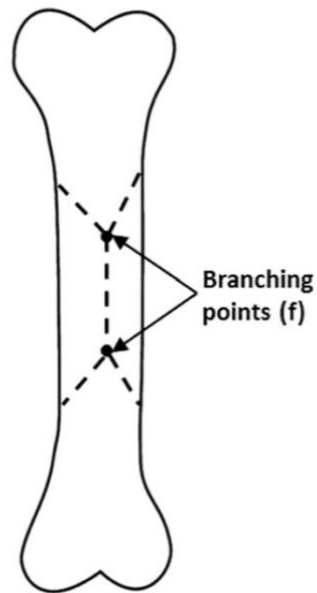


Ilustración 10. Puntos de ramificación

- Fragmentos de viruta: Es la parte ósea faltante en un punto de impacto. Las características relacionadas son la presencia y la longitud de los fragmentos en mm principalmente.
- Número de líneas
- Ubicación de la línea
- Curvatura de la línea: número de veces que la línea se desvía de su curso.

2.5. Herramientas para la edición de patrones de fractura

En lo referente a las herramientas existentes para la generación de un patrón de fractura, se puede concluir en que con cualquier editor de dibujado podríamos obtener una imagen de un patrón de fractura, pero esto no es lo más óptimo debido a que los editores de dibujo no tienen como objetivo el dibujado de un patrón de fractura, por lo que no están adaptados a ello. Además, no se pueden visualizar otros patrones o exportar el patrón mediante un fichero de texto en el que tengamos todos los elementos de dibujado, así como tampoco se podría aplicar ningún tipo de validación sobre esta fractura, ya que únicamente tendríamos una imagen con elementos dibujados, a los que no tendríamos acceso y que no tendrían ningún tipo de relación entre ellos.

Por este motivo, es fundamental la creación de un editor específico para tales objetivos. Este Trabajo de Fin de Grado, se va a encargar de la realización de un editor dedicado exclusivamente a la creación y validación de patrones de fractura, tanto creados en el editor como importados de otros sistemas, creando así la primera herramienta software capaz de realizar tal trabajo.

3. Diseño del prototipo

Este capítulo se va a dedicar al diseño del prototipo incluyendo la metodología seguida, las fases de la ingeniería del software requeridas y detalladas, así como el entorno de trabajo elegido.

3.1. Metodología de generación del prototipo

La metodología empleada ha sido metodología SCRUM [1], en la que se trabaja en iteraciones o Sprint. Se ha elegido esta metodología porque es flexible, basada en entregas parciales y regulares del producto final en base al valor que ofrecen los clientes. Entre sus principales características destaca el hecho de que sea un desarrollo incremental en lugar de la clásica planificación del desarrollo completo de un producto o servicio. Sus equipos de trabajo se caracterizan por ser auto-organizados. Y se centra en el producto final y en la calidad del mismo. Además, en la metodología Scrum se solapan diferentes

fases de desarrollo, en lugar de llevar a cabo una planificación secuencial o de cascada.

El enfoque iterativo está basado en equipos. Este enfoque enfatiza la entrega rápida de una aplicación en componentes funcionales completos. Cada Sprint tiene una duración definida con una lista continua de entregables, planificada al principio del Sprint. Los entregables son priorizados por el valor del negocio, determinado por el cliente. Si no se puede completar todo el trabajo planificado para el sprint, se vuelven a priorizar las tareas y se incluyen en el siguiente sprint.

Gracias a esta metodología se puede integrar al cliente completamente e ir haciéndolo participe en todo el proyecto, ya que a medida que el trabajo se va completando puede ir siendo revisado por el equipo del proyecto y el cliente, pudiendo detectar posibles cambios en el trabajo, tomar decisiones o incluso ir lanzando versiones básicas del software que podrá ser completado posteriormente en sucesivas iteraciones.

La metodología Scrum considera cinco fases de trabajo, definidas por tiempos de ejecución.

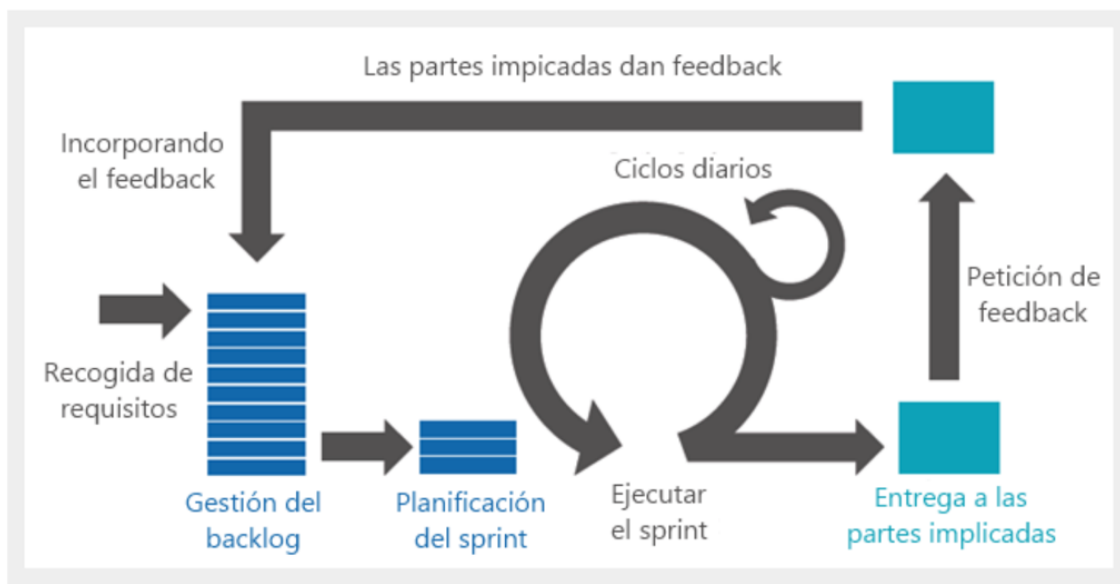


Ilustración 11. Metodología Scrum. Fases

a) Recogida de requisitos

Comienza con la creación de la lista de objetivos o requisitos priorizada, que actúa como plan del proyecto y que es entregada por el cliente o dueño del producto al equipo. Esta lista representa la visión y las expectativas del cliente respecto a los objetivos y entregas del producto o proyecto.

Esta etapa correspondería con la fase de planificación de un proyecto en un marco no ágil de trabajo.

b) Gestión de backlog

Comprende el conjunto de funcionalidades y tareas a realizar. Para cada objetivo/requisito se indica el valor que aporta al cliente, para así priorizar.

c) Planificación del Sprint

Un sprint es una unidad de trabajo que agrupa un conjunto de tareas en un periodo de tiempo. La primera iteración es de planificación y está compuesta por dos partes:

Selección de requisitos: Es la iteración entre cliente y equipo, el momento en que el equipo pregunta al cliente las dudas que surgen y se seleccionan los requisitos más prioritarios que se comprometen a completar en la iteración. Tiene una duración máxima de cuatro horas.

Planificación de la iteración: Se elabora la lista de tareas o acciones necesarias para desarrollar los requisitos a los que se han comprometido. La estimación de esfuerzo se hace de manera conjunta, siempre con el scrum master como facilitador, y los miembros del equipo se asignan a sí mismos las tareas. La duración de este ejercicio no debe superar las cuatro horas.

d) Ejecución del Sprint

En esta metodología, un proyecto se ejecuta en bloques temporales cortos y fijos, llamados sprint, que son iteraciones de 2 semanas de duración. Si se sobrepasa este tiempo, como máximo puede durar 4 semanas.

Cada día se hace una reunión diaria de no más de 15 minutos en la que se explica que se ha hecho, que se va a hacer y que impedimentos se tienen o se van a tener.

e) Inspección e iteración

Cuando se acaba la iteración se realiza una revisión compuesta por:

- **Sprint Review:** el equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado y el cliente lo revisa.
- **Sprint Retrospective:** el equipo analiza como ha sido su forma de trabajar y cuales son los problemas que pueden impedirle trabajar adecuadamente, enfocando el proceso a la mejora continua del equipo.

Una vez acabada esta fase se procede a la entrega del producto para pruebas.

3.2. Entorno de trabajo

En este apartado se va a describir el entorno de trabajo a usar, indicando también el entorno de trabajo con el que se empezó a trabajar y el motivo por el cuál se descartó.

3.2.1. Entorno de trabajo inicial

Para la realización del proyecto se ha creado un entorno de trabajo que consta de una serie de elementos, tales como:

- Windows 10 como sistema operativo
- Visual Studio Enterprise 2017, en concreto el módulo de desarrollo para escritorio de C++
- Librerías PCL [2]
- Git
- Cmake 3.11.0 [3]

La elección de dichas herramientas viene dada por la búsqueda de compatibilidad con otras investigaciones desarrolladas en el grupo de trabajo de la Universidad de Jaén en este campo.

Para la configuración del entorno de trabajo he comenzado instalando el sistema operativo anteriormente mencionado en el ordenador. Para ello, he creado una máquina virtual con 100 GB de disco duro y 8GB de RAM, teniendo así recursos suficientes para trabajar de forma correcta en la ejecución del proyecto.

Como herramienta de desarrollo he usado Visual Studio Enterprise 2017 con C++ como lenguaje de programación. Tras lo cuál he creado una solución software en la cuál he realizado la implementación del proyecto.

Una vez instalado el sistema operativo y creado el proyecto de Visual Studio, he procedido a compilar las bibliotecas de PCL y sus dependencias mediante cMake.

En primer lugar, mediante la interfaz gráfica de cMake he realizado la configuración de la siguiente forma:

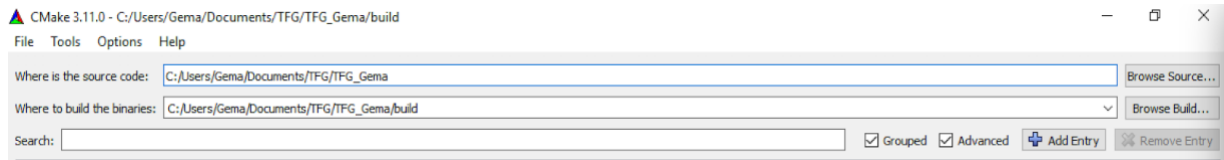


Ilustración 12. Configuración cMake

Tras hacer esto, en una primera instancia aparecen varios errores. El primero de ellos dice que no encuentra PCL, para solucionarlo simplemente se le configura manualmente la ruta en la que está la carpeta con la biblioteca de PCL. Una vez hecho esto, aparece todo lo que se tiene que compilar. Si se vuelve a ejecutar, aparecen nuevos errores.

El segundo de ellos hace referencia a que no se encuentra VTK, que se soluciona poniendo la ruta manualmente en la interfaz de cMake, quedando así la configuración de esa parte:

Name	Value
Ungrouped Entries	
EIGEN_INCLUDE_DIRS	C:/Users/Gema/Documents/TFG/librerias/PCL1.8.1/3rdParty/Eigen/eigen3
PKG_CONFIG_EXECUTABLE	PKG_CONFIG_EXECUTABLE-NOTFOUND
VTK_DIR	C:/Users/Gema/Documents/TFG/librerias/PCL1.8.1/3rdParty/VTK/lib/cmake/vtk-8.1

Ilustración 13. Configuración de cMake (II)

Otro de los errores que han aparecido es de OpenNI2, que se soluciona yendo a la carpeta en la que están las bibliotecas y siguiendo la ruta relativa './PCL1.8.1/3rdParty/OpenNI2', ejecutamos el archivo que aparece ahí, llamado OpenNI-Windows-x64-2.2.msi

Una vez hecho esto, al darle a configurar o a generar en la interfaz de cMake, logra compilarse con éxito.

El siguiente paso para la configuración correcta es la creación de las variables de entorno del sistema.

En el apartado variables de sistema, editamos la variable PATH, añadiendo las siguientes líneas de configuración:

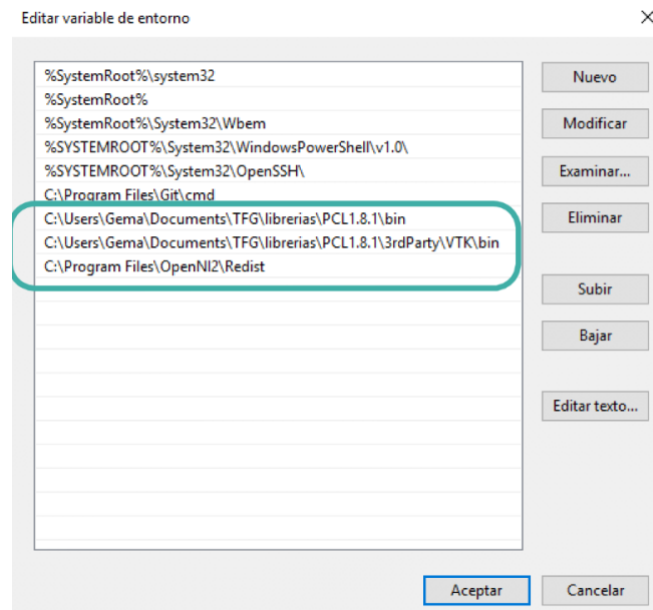


Ilustración 14. Configuración de las variables de entorno

Finalmente, en Visual Studio, con el proyecto abierto, se va a la configuración del proyecto para establecer que se comience ejecutando la carpeta de mi código fuente y no la carpeta de las bibliotecas que viene por defecto.

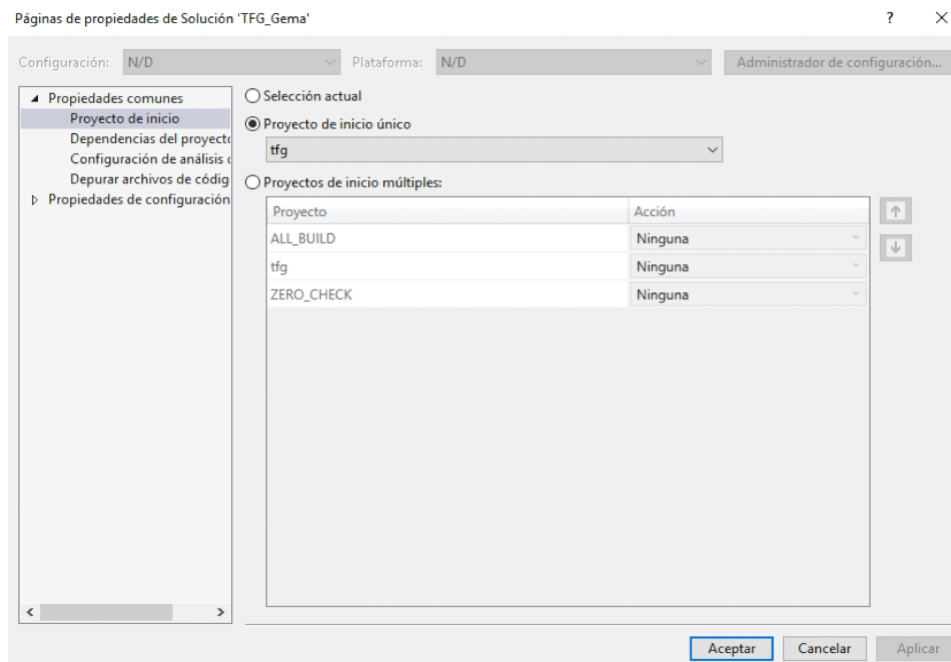


Ilustración 15. Configuración de Visual Studio

3.2.2. Justificación de cambio de entorno de trabajo

Tras probar el primer entorno de trabajo, que era el que se estaba usando en el grupo de trabajo de la Universidad de Jaén, se encontraron diversas limitaciones en la biblioteca PCLPainter2D, ya que no se podría realizar la mayoría de las cosas que debía hacer en el proyecto, como por ejemplo integrar PCLPainter2D con eventos de ratón para el dibujado de las fracturas, por ello, tras mucho investigar y viendo que no podía ser funcional, decidí cambiar de entorno de trabajo.

En la elección del nuevo entorno de trabajo, que se encuentra detallado en el siguiente apartado, me he decantado por el entorno que mejor se adapta a las necesidades del prototipo a implementar permitiendo la interacción del ratón en el dibujado y el manejo de primitivas básicas de dibujado como son las líneas y los puntos de forma sencilla.

3.2.3. Entorno de trabajo definitivo

Tras el cambio de entorno de trabajo, al contar con un ordenador Macos, decidí usarlo y ahorrarme el paso de tener que montar una máquina virtual como en el entorno inicial y tener que trabajar en un entorno virtualizado con todo lo que conlleva.

En primer lugar, para la realización del proyecto se ha creado un entorno de trabajo que consta de:

- Macos como sistema operativo
- xCode, en concreto el módulo de desarrollo de C++
- Librerías de gooCanvas para dibujar [4]
- Librería de GTK+, para la interfaz [5]
- Aplicación Glade [6] como apoyo de GTK+ para hacer la interfaz de una forma más intuitiva
- Git para tener un control de versiones.

En la configuración del entorno de trabajo he elegido macOS High Sierra, que cuenta con 16 GB de RAM y 512 GB de almacenamiento, aunque con menos capacidad el proyecto funcionará de igual forma.

Como herramienta de desarrollo he usado xCode 10.1 con C++ como lenguaje de programación. Tras hacer esto he creado una solución software en la cuál he realizado la implementación del proyecto.

Una vez instalado el sistema operativo y creado el proyecto de xCode, he procedido a instalar y vincular las bibliotecas con el proyecto.

GTK+

En primer lugar, he descargado GTK+ de <https://www.gtk.org/download/macos.php> y he ejecutado este archivo mediante la línea de comandos, dándole previamente permisos de ejecución.

```
MBP-de-Gema:Desktop gema$ sudo chmod 777 /Users/gema/Desktop/gtk-osx-build-setup.sh
MBP-de-Gema:Desktop gema$ /Users/gema/Desktop/gtk-osx-build-setup.sh
Checking out jhbuild (7c8d34736c3804) from git...
```

Ilustración 16. Configuración GTK+

Tras hacer esto, se ha vinculado la librería con el proyecto de xCode, así como todas sus dependencias.

Finalmente, he instalado la biblioteca gooCanvas desde aquí: “git clone <https://gitlab.gnome.org/GNOME/goocanvas.git>” y la he vinculado al proyecto añadiendo sus dependencias en el apartado correspondiente en el xCode (Ilustración 8, 9 ,10).

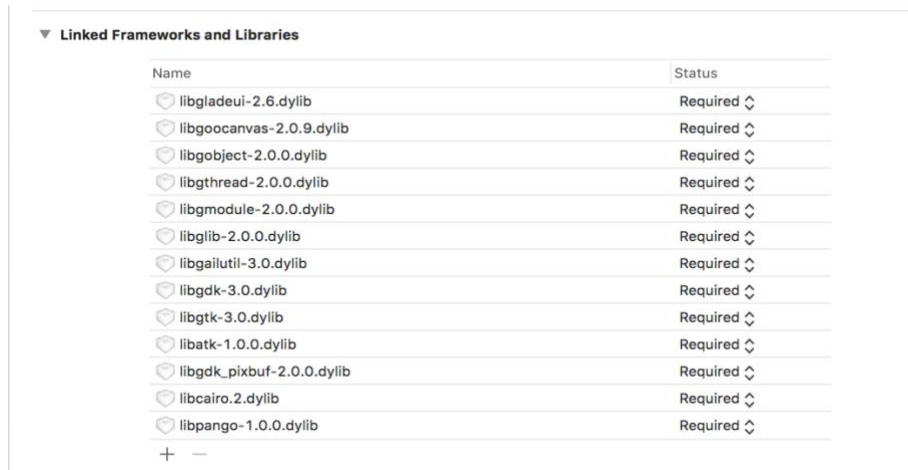


Ilustración 17. Hacer el link de las librerías

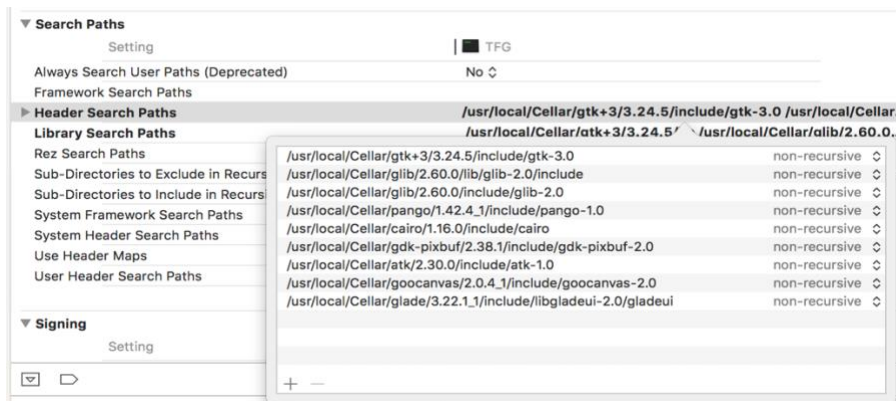


Ilustración 18. Linkear librerías (II)

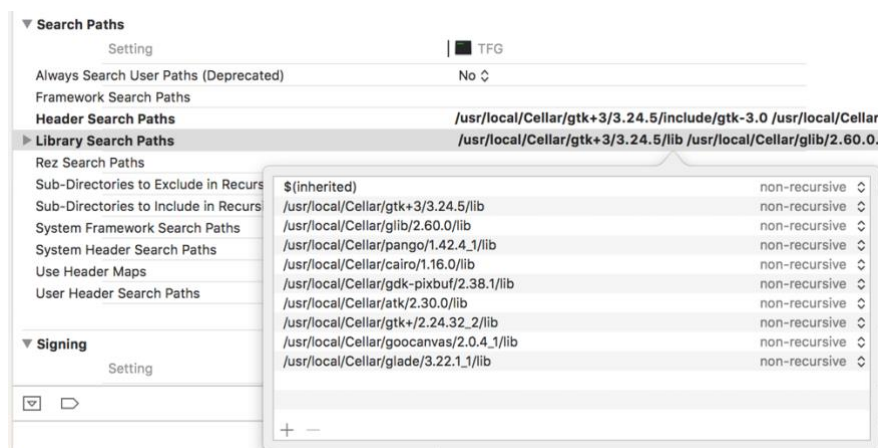


Ilustración 19. Linkear librerías (III)

También he incluido Eigen [7], una biblioteca matemática que usaré de apoyo para cálculos relacionados con la geometría de las líneas de fractura.

Finalmente, he instalado Glade <https://glade.gnome.org>, para un manejo más intuitivo de los componentes de la interfaz, con *brew install glade* en la terminal.

En cuanto a las tecnologías usadas para la implementación, se ha elegido C++ como lenguaje de programación al ser un lenguaje robusto y versátil por lo que lo he considerado indicado para la realización del proyecto. Además, es un lenguaje flexible, está orientado a objetos y es muy potente en la creación de sistemas complejos.

Para la implementación de la parte visual he decidido usar GooCanvas con GTK+. GTK es un acrónimo para GIMP ToolKit y es una plataforma para la creación de entornos gráficos muy popular para Linux/Unix, aunque también se puede usar en otros sistemas operativos.

GTK se basa en varias bibliotecas que nos permiten crear la parte gráfica de nuestra aplicación, las más relevantes para mi proyecto son:

GTK: que es la biblioteca que contiene los objetos y funciones para crear la interfaz de usuario (widget, botones, menús, etiquetas, etc)

- GDK: biblioteca que actúa como intermediario entre los gráficos de bajo nivel y los de alto nivel.
- Cairo: biblioteca de renderizado avanzado de controles de aplicación

Por otro lado, GooCanvas es un lienzo que se usa Cairo como widget en GTK.

3.3. Ingeniería Software

Este capítulo va a englobar todo lo relativo al proceso de ingeniería del software, incluyendo sus distintas fases [8].

3.3.1. Planificación

La planificación de un proyecto, así como la estimación de las diferentes tareas son puntos críticos para el éxito de un proyecto. La planificación es difícil y la mayoría de las veces suele incumplirse, pero un buen proceso de planificación consigue que se reduzca el riesgo de fracaso del proyecto y la mejora a la hora de tomar decisiones.

Para la consecución de los objetivos se ha establecido una planificación temporal mostrada que detalla el tiempo invertido en cada una de las partes, considerando semanas de 5 días y 4 horas de trabajo.

Tarea	Temporización
Estudio bibliográfico y documentación	1 semana
Diseño de la aplicación e interfaz	2 semanas
Diseño de formatos de exportación	3/2 semanas
Implementación	7 semanas
Pruebas de patrones de fractura	1 semana
Validación de resultados	2 semanas
Generación de patrones de fractura	3/2 semanas
Elaboración de documentación y publicaciones	2 semanas
Total	18 semanas

Tabla 1. Planificación temporal

En la primera fase, se tratará de obtener toda la información necesaria sobre fracturas y las distintas formas de generación de las mismas para tener una visión general acerca de los patrones de fracturas. En esta fase se comenzarán a analizar los distintos parámetros necesarios para la creación de fracturas válidas, así como que tipos de fracturas o elementos deberá de tener el editor. Una vez hecho esto, en la siguiente fase, habrá que realizar el diseño

de la interfaz que recoja todos los parámetros necesarios. Por otro lado, la tercera fase se corresponde con la fase en la que se diseñará el formato de exportación de los datos para hacer compatible la exportación de las fracturas de este sistema con otros sistemas. Tras hacer esto, se procederá a la fase de la implementación de la herramienta seguida de las pruebas de patrones de fractura. Una vez realizadas estas pruebas, se procederá a la validación de los resultados tanto con un validador automático que se creará en la propia herramienta como con un experto que valide los patrones de fractura. Finalmente, se generarán los distintos patrones de fractura y se exportarán. En cuanto a la documentación, aunque esté indicada en el final del proyecto, es una tarea que se realizará de forma paralela durante la duración de todo el proyecto.

Para simplificar la visualización de las tareas en el tiempo, se ha representado la temporización junto con las tareas mediante un diagrama de Gantt [9].

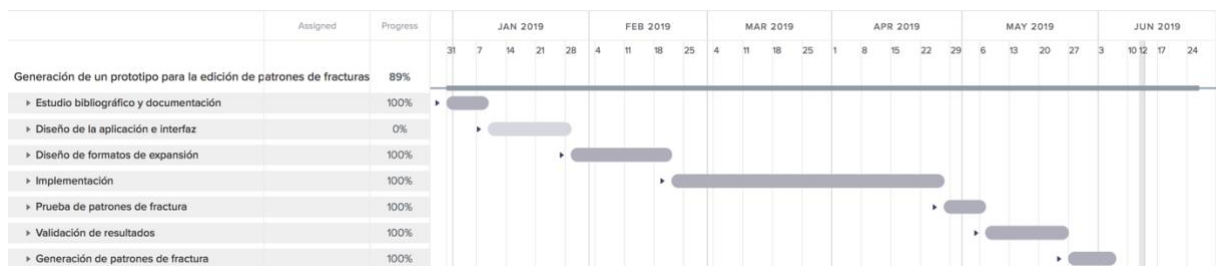


Ilustración 20. Diagrama de Gantt

3.3.2. Estimación de costes.

En cuanto a los costes derivados de la realización del proyecto es necesario tener en cuenta los costes asociados al software, al hardware y al personal.

- Hardware

Únicamente se ha necesitado un ordenador para programar. Dicho equipo tiene una duración media entre los 5 y 10 años, considerando una duración media de 5 años y el coste total del equipo de 2700 €, la cantidad a amortizar durante los 4 meses de desarrollo es de 180€.

Hardware	Precio
MacBook Pro (15-inch, 2017)	180 €
Total	180 €

Tabla 2. Costes hardware

- Software

Debido a que el software usado es gratuito y las tecnologías usadas durante el desarrollo son de código abierto o gratuitas, el coste global es 0 €.

Software	Precio
Xcode	0 €
Glade	0 €
Total	0 €

Tabla 3. Costes software

- Gastos derivados

En los gastos derivados se incluyen todo tipo de gastos, tales como gastos en electricidad y cuota de internet.

Costes derivados	Precio
Electricidad	6,83 €/mes x 4 meses = 27,33 €
Internet	40 €/ mes x 4 meses = 160 €
Total	187, 33 €

Tabla 4. Costes derivados

- Gastos en trabajadores

La elaboración de todo proyecto necesita de distintos roles que hagan que dicho proyecto llegue al éxito. En este proyecto se va a necesitar un equipo compuesto por los siguientes roles:

- Jefe de proyecto

Será el encargado de planificar y de coordinar todas las tareas desarrolladas en el proyecto, así como de gestionar los recursos económicos y materiales para obtener los resultados esperados en los plazos previstos y con la calidad necesaria.

- Analista

Es la persona encargada de establecer los requisitos técnicos del proyecto.

- Diseñador

Es la persona encargada de la realización del diseño para la aplicación.

- Programador

Se encarga de la implementación de la aplicación

- Encargado de realizar los test

Se encarga de realizar las distintas pruebas del prototipo para comprobar que la aplicación funciona correctamente y detectar posibles fallos.

Este proyecto va a ser realizado por únicamente por mí, por lo que asumiré todos los roles mencionados con anterioridad. Teniendo en cuenta los salarios [10] medios de los distintos roles y las retenciones de seguridad social, así como el grado de dedicación de horas a cada una de las tareas. En cuanto a la duración, hace referencia a un mes en el que se trabajan 5 días a la semana un total de 4 horas al día. Los gastos en recursos humanos vienen detallados en la siguiente tabla.

Tabla 5. Gastos en recursos humanos

Rol	Sueldo neto	Retención	Dedicación	Duración	Total
Analista	24700 €	13,96 %	100 %	1 mes	820.15 €
Programador	22600 €	13,96 %	100 %	3 meses	2316.15 €
Tester	23000 €	13.96 %	50 %	2 meses	772.05 €
Diseñador	18500 €	13.96 %	100 %	medio mes	319.275 €
Jefe de proyecto	31500 €	13.96 %	25 %	4 meses y medio	1257.15 €
TOTAL					5484,775 €

- Gasto total

Teniendo en cuenta todos los aspectos que hemos mencionado a lo largo de este apartado se puede concluir que el presupuesto del proyecto asciende a 8372,105€ para cubrir con los gastos del proyecto al completo.

3.3.3. Iteraciones

Las iteraciones realizadas en el proyecto no siempre han tenido la misma duración, ya que algunas de las tareas requerían de menor tiempo de dedicación y otras de menos. La duración de cada una de las iteraciones viene dada por la tabla de planificación temporal, a excepción de las iteraciones correspondientes a la implementación, ya que la implementación se ha dividido en distintas iteraciones para así hacer una planificación más detallada y que sea más sencilla de llevar a cabo.

A lo largo de todas las iteraciones se irá realizando la documentación del proyecto de forma paralela, aunque no se mencione.

3.3.3.1. 1ª Iteración. Estudio bibliográfico y documentación

En este apartado se expone todo lo necesario para la realización de la primera iteración.

- **Análisis de requisitos**

En esta iteración no se dispone de fase de análisis de requisitos.

- **Diseño**

En esta iteración no se dispone de fase de diseño.

- **Pruebas**

En esta iteración no se dispone de fase de pruebas.

- **Proceso**

Debido a que es la primera vez que se aborda un tema relacionado con fracturas, esta iteración va a ser dedicada a la búsqueda de información sobre patrones de fractura y al análisis y estudio exhaustivo de la misma con el fin de tener los conocimientos necesarios y suficientes para la realización del proyecto.

El objetivo principal de esta fase es conocer que es un patrón de fractura, los distintos tipos y qué parámetros los definen.

3.3.3.2. 2ª Iteración. Diseño de la aplicación e interfaz.

Una vez tenemos los conocimientos necesarios acerca de los patrones de fractura y de su modo de representación, es necesario pasar al diseño de la interfaz, puesto que va a ser una parte fundamental en la aplicación.

- Análisis de requisitos

Requisitos funcionales

- Debe tener un área de dibujo que represente un hueso desplegado, con líneas divisorias que separen cada uno de los aspectos del hueso.
- Debe permitir dibujar desprendimientos de fragmento de hueso.
- Debe permitir que la fractura no tenga desprendimiento de fragmento.
- Debe permitir dibujar líneas.
- Debe permitir ajustar los parámetros de un fragmento: tipo, tamaño, rotación.
- Debe permitir ajustar las características de una línea: color, tipo y grosor.
- Debe permitir la exportación e importación de patrones de fractura.
- Debe permitir cambiar el grosor del hueso.
- Debe permitir poder validar patrones.

Requisitos no funcionales: no hay requisitos no funcionales.

- Diseño

Como en todo proyecto, antes de comenzar a implementar nada, hay que diseñar el prototipo a realizar para tener una visión general de lo que se quiere hacer y abordar esa tarea de la forma más fácil y eficaz posible.

La propuesta inicial del prototipo consta de un canvas de dibujado a la izquierda y un selector de características de la línea de fractura y del segmento a la derecha y una pestaña para la exportación e importación.

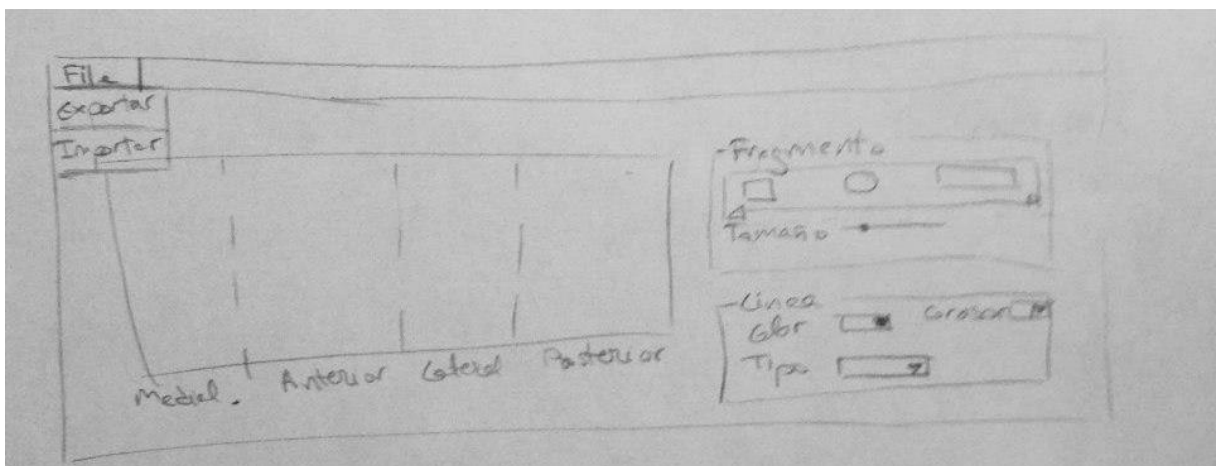


Ilustración 21. Storyboard aplicación

Tras el diseño inicial, se le ha añadido un “tab” en el que están las opciones de “Archivo”, “Editar” y “Ayuda”, “Ver” y “Validar” para una mejor gestión de la aplicación, quedando un prototipo de interfaz como se muestra en la siguiente imagen.

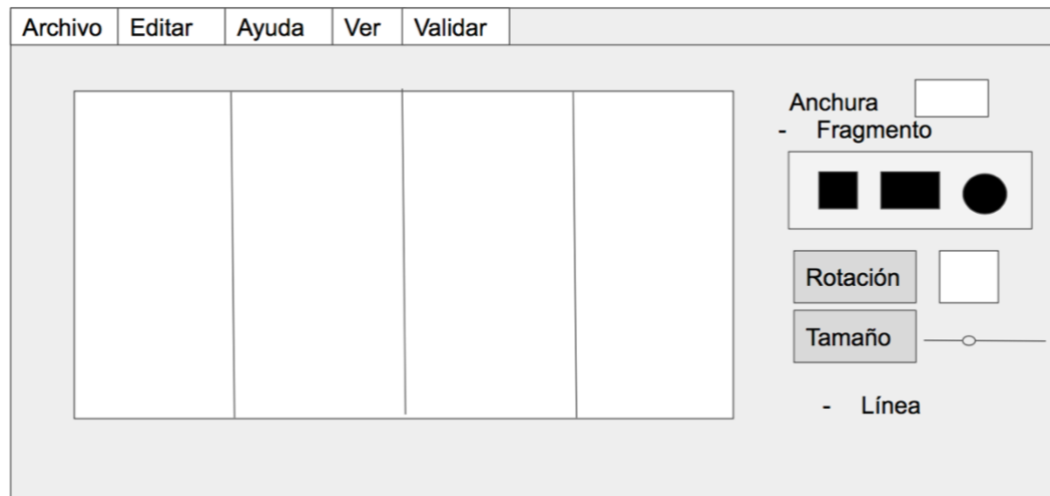


Ilustración 22. Prototipo a papel 1

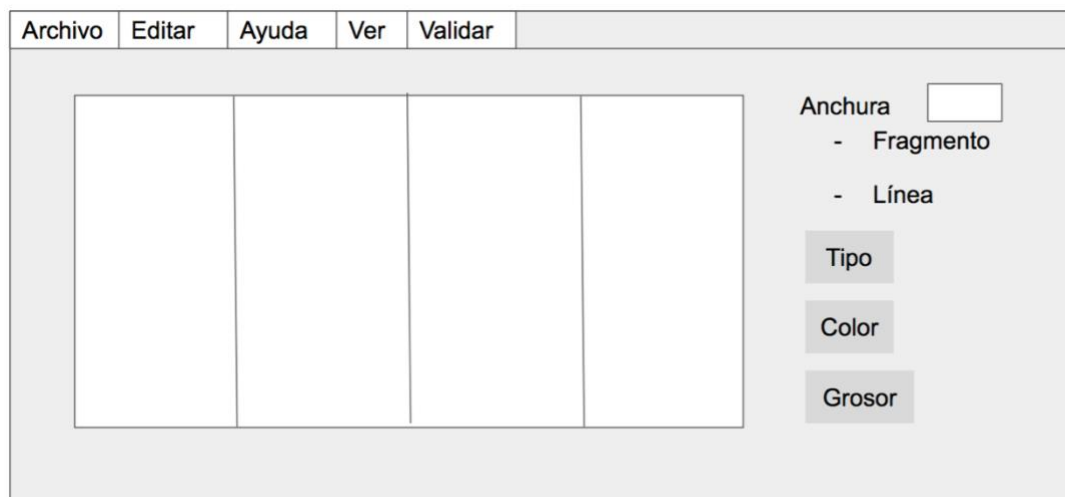


Ilustración 23. Prototipo a papel 1

- Pruebas

En esta iteración no hay fase de pruebas.

3.3.3.3. 3ª Iteración. Diseño de los formatos de exportación.

En esta iteración se realizará el diseño del formato de exportación del patrón de fractura. Para ellos se tendrán en cuenta todos los valores necesarios para la replicación de una fractura en otro sistema.

- Análisis de requisitos

Requisitos funcionales

- Contenga el fragmento desprendido si lo hay y su tipo
- Contenga todas las líneas generadas en la fractura y su tipo
- Contenga el color, tipo y grosor de líneas
- Contenga las características del fragmento (color de relleno y color de contorno)

Requisitos no funcionales:

- Debe ser compatible con otros sistemas. Necesariamente debe serlo con el generador de fracturas aleatoria presente en el Grupo de Investigación de la Universidad de Jaén.

- Diseño

Se ha considerado necesario que el formato de exportación sea pf2d, debido a que es el formato usado por los otros sistemas para la importación.

En cuanto al diseño del formato de exportación, se pueden encontrar el patrón de exportación y una explicación exhaustiva en el apartado [3.4.](#), debido a que al ser un aspecto de interés en la aplicación y al tener que ser compatible con otros sistemas, bajo mi punto de vista merece un apartado dedicado a él para que así sea más localizable y fácil de consultar.

- Pruebas

Las pruebas realizadas tratan de la creación de un fichero de ejemplo de exportación y probarlo en otros sistemas para comprobar su correcta adaptación.

3.3.3.4. 4ª Iteración. Implementación → Desarrollo de la interfaz.

En esta iteración se implementará todo lo relativo a la interfaz, teniendo al acabar dicha iteración un prototipo con una interfaz acabada visualmente, así como responderá a los distintos eventos, pero que no tendrá implementada la funcionalidad real de cada una de sus partes.

- Requisitos

Requisitos funcionales:

- Debe tener un área de dibujo que represente un hueso desplegado, con líneas divisorias que separen cada uno de los aspectos del hueso.
- Debe permitir dibujar desprendimientos de fragmento de hueso.
- Debe permitir que la fractura no tenga desprendimiento de fragmento.
- Debe permitir dibujar líneas.
- Debe permitir ajustar los parámetros de un fragmento: tipo, tamaño, rotación.
- Debe permitir ajustar las características de una línea: color, tipo y grosor.
- Debe permitir la exportación e importación de patrones de fractura.
- Debe permitir cambiar el grosor del hueso.
- Debe permitir poder validar patrones.

Requisitos no funcionales

- Debe ser fluido, usable y fiable.

Los requisitos indicados se plasman en los casos de uso detallados en el siguiente diagrama:

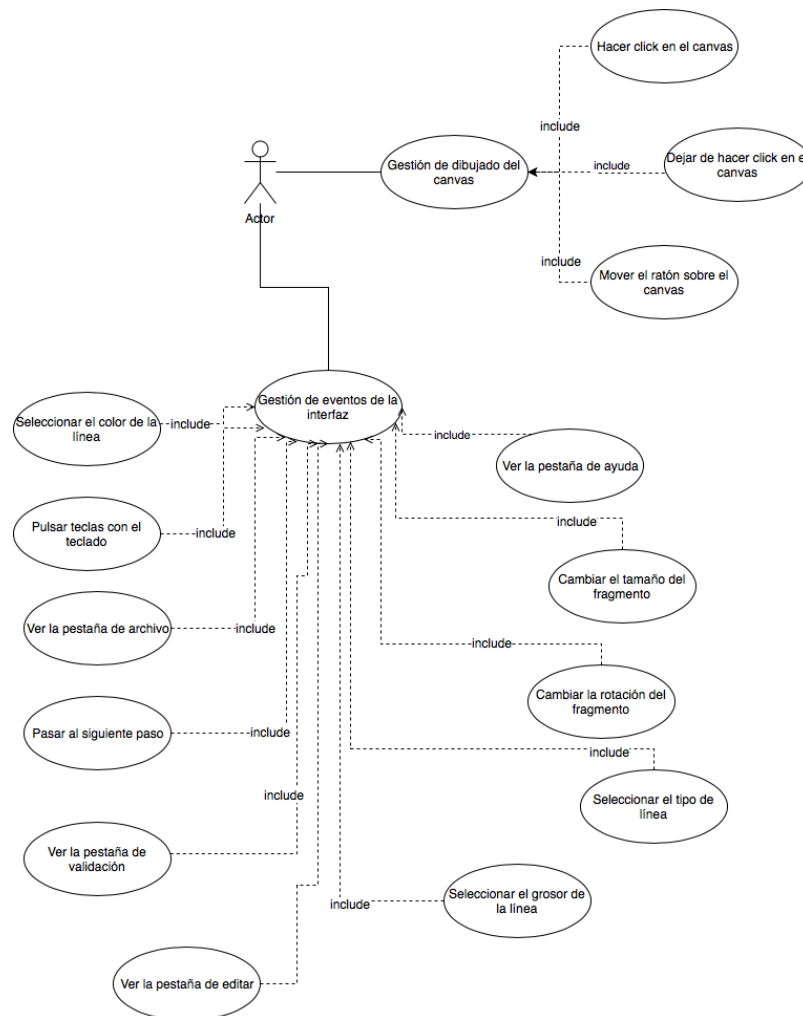


Ilustración 24. Diagrama de casos de uso de la interfaz

- Diseño

Debido a que la interfaz gráfica es una parte de la aplicación ajena al funcionamiento de la aplicación, ya que se pueden tratar como dos elementos independientes, se ha decidido encapsular todo lo relativo a la implementación de la interfaz en una clase denominada “Interfaz”, que será la encargada de la creación de la ventana de la aplicación y de situar en ella todos los elementos visuales definidos en la misma, así como de controlar los eventos de la aplicación. Para dicha tarea de detección de

eventos en la aplicación, se ha decidido crear otra clase llamada “CallbacksFunctions”, que contenga todos los callbacks que puedan ser invocados desde la interfaz de la aplicación.

Por otro lado, también he considerado crear otra clase, la clase Painter, que será la encargada de contener el canvas y todo lo asociado a la misma. ”. Dicha clase se encarga de todo lo referente al dibujado de un patrón de fractura y los métodos de esta clase son invocados desde cualquier parte de la aplicación, ya que estará implementado en forma de “singleton” para que solo pueda haber una instancia activa de ese tipo por aplicación.

Básicamente todos los casos de uso descritos en el diagrama anterior son producidos por eventos de ratón o teclado, por lo que estarán en la clase CallbacksFunctions y se asociarán a los distintos elementos de la interfaz, tanto los generados mediante gtk, que son los botones y distintos elementos de la interfaz propiamente dichos, como al canvas de dibujado, que usará la biblioteca gooCanvas y se enlazará al propio canvas los distintos eventos asociados al dibujado como por ejemplo los casos de uso de pinchar, soltar y deslizar por el canvas.

A continuación, se muestra el diagrama UML de las clases.

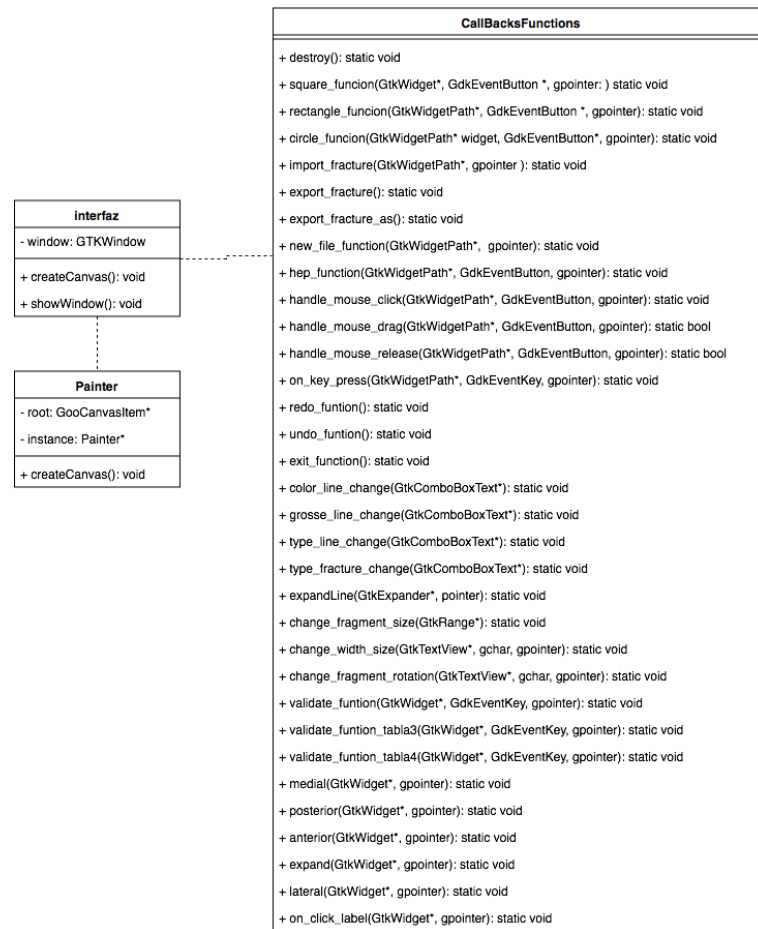


Ilustración 25. UML diseño de la interfaz y eventos necesarios

En el diagrama UML se puede observar que en la clase Interfaz tenemos la ventana de la aplicación, que una vez creada creará el canvas mediante el método de creación del canvas de dibujado, mediante la clase Painter.

Por otro lado, podemos observar que han sido estudiados los distintos métodos que serán necesarios para responder a todos los eventos producidos en la interfaz de la aplicación.

- Pruebas

En esta iteración las pruebas han consistido en el uso de la aplicación para comprobar que cada parte de la interfaz tenía asociado su evento correspondiente, aunque no fuese funcional.

El estado de la interfaz del prototipo tras esta iteración es el siguiente:



Ilustración 26. Interfaz del prototipo

3.3.3.5. 5ª Iteración. Implementación → Dibujado de líneas.

Esta iteración engloba la base del proyecto, ya que se encarga de la parte esencial del desarrollo, que es el dibujado del patrón de fractura. Se implementará todo lo necesario para que un usuario pueda dibujar sobre el lienzo mediante el ratón, permitiendo distintos tipos de fractura. De igual forma, se buscará tener como resultado un dibujado de líneas de fractura con ciertas restricciones generales de todos los tipos de fracturas y un orden de dibujado para asegurar dentro de los límites de la herramienta que el usuario haga un dibujado del patrón de fractura lo más correcto posible.

- Requisitos

Requisitos funcionales

- Debe permitir dibujar todos los tipos de líneas.

- Debe permitir que las ramificaciones se entrecrucen.
- Debe prohibir que las líneas oblicuas se corten.
- Debe permitir que una línea oblicua salga por un lado del canvas de dibujado y se pueda continuar dibujando por el lado opuesto, dado que un hueso es cilíndrico.
- Las líneas longitudinales tienen que estar en contacto con otras líneas, si no es el aspecto impactado, o con el fragmento o punto de fractura si lo es, para poderse dibujar.
- Debe permitirse dibujar un fragmento
- Debe permitirse no dibujar fragmento y que solo haya punto de fractura

Requisitos no funcionales

- Debe ser un dibujado fluido
- Debe ser una herramienta usable y fiable.

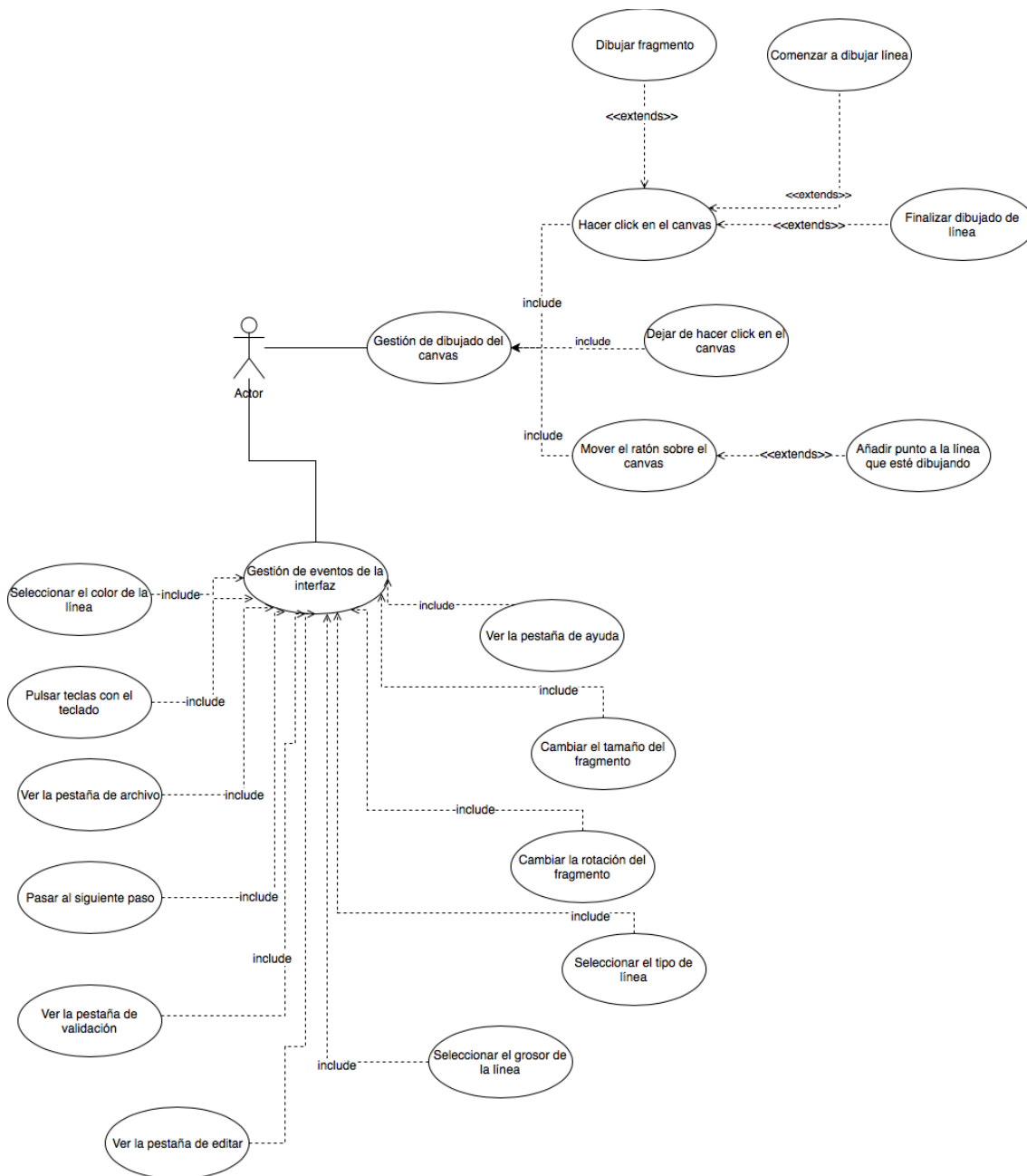


Ilustración 27. Diagrama de casos de uso del dibujo de la línea

- Diseño

En esta iteración se va a abordar el dibujo sobre el canvas. Por ello, habrá que tener en cuenta las distintas restricciones en el dibujo tanto de las líneas como del fragmento, así como velar por la fluidez de la aplicación.

Para esta última tarea, se ha decidido usar como estructura de datos para un control del “canvas” de dibujo preciso, tanto en eficiencia como

en precisión de selección un grid. Por ello, he decidido crear la clase “Grid”, que encapsula todo lo referente al mismo y al manejo de datos en él, como el borrado de datos, intersección, localización, etc.

Para controlar los datos contenidos en el grid, se ha decido crear la clase “CanvasItem”, que se encarga de almacenar todo lo relacionado con la línea dibujado o fragmento, como por ejemplo los puntos que lo forman, su centro, ramificaciones que salen de él, ...

Finalmente se ha creado otra clase, “GeometryLib”, que contiene todo lo relativo a cálculos geométricos para así poder llevar acabo ciertas restricciones como que las líneas no se crucen o comprobar si se está pinchando sobre una línea ya existente para comenzar a dibujar.

A continuación, se muestra el diagrama de clases empleado para el desarrollo de esta iteración.



- Pruebas

En esta fase las pruebas son mucho más exhaustivas, ya que se procederá a usar la herramienta e intentar todas las posibles combinaciones de dibujados de líneas para probar y verificar que la herramienta realiza un dibujo correcto y fiable de las líneas.

3.3.3.6. 6ª Iteración. Implementación → Funcionalidad completa del sistema.

En esta iteración se completará toda la funcionalidad de la herramienta, como por ejemplo la exportación, importación, deshacer/rehacer dibujo, crear nuevo lienzo en blanco, cambio de color de líneas de fractura y de estilo, rotación del fragmento de viruta y escalado, etc.).

- Requisitos

Requisitos funcionales:

- Se deberá permitir deshacer las operaciones tanto mediante la pestaña de edición como mediante teclado.
- Se deberá permitir rehacer operaciones tanto mediante la pestaña de edición como mediante teclado.
- Se deberá permitir realizar la exportación de un patrón de fractura.
- Se deberá permitir realizar la importación de un patrón de fractura de otro sistema y del propio.
- Se deberá permitir crear un nuevo lienzo de dibujo
- Se deberá permitir cambiar las características del fragmento.
- Se deberá permitir cambiar las características de dibujo de una línea, permitiendo tener en la aplicación líneas de distintos grosores o tipos.
- Se deberá permitir que se cambie el tamaño del canvas de dibujo.

- Se deberá permitir que se pueda navegar entre los pasos de la interfaz para poder saltarse los pasos que no sean requeridos.
- Deberá ser compatible con otros archivos exportados a otras herramientas, así como también deberá ser compatible con los importados a la propia.

Requisitos no funcionales:

- Deberá ser una herramienta fiable y fluida.

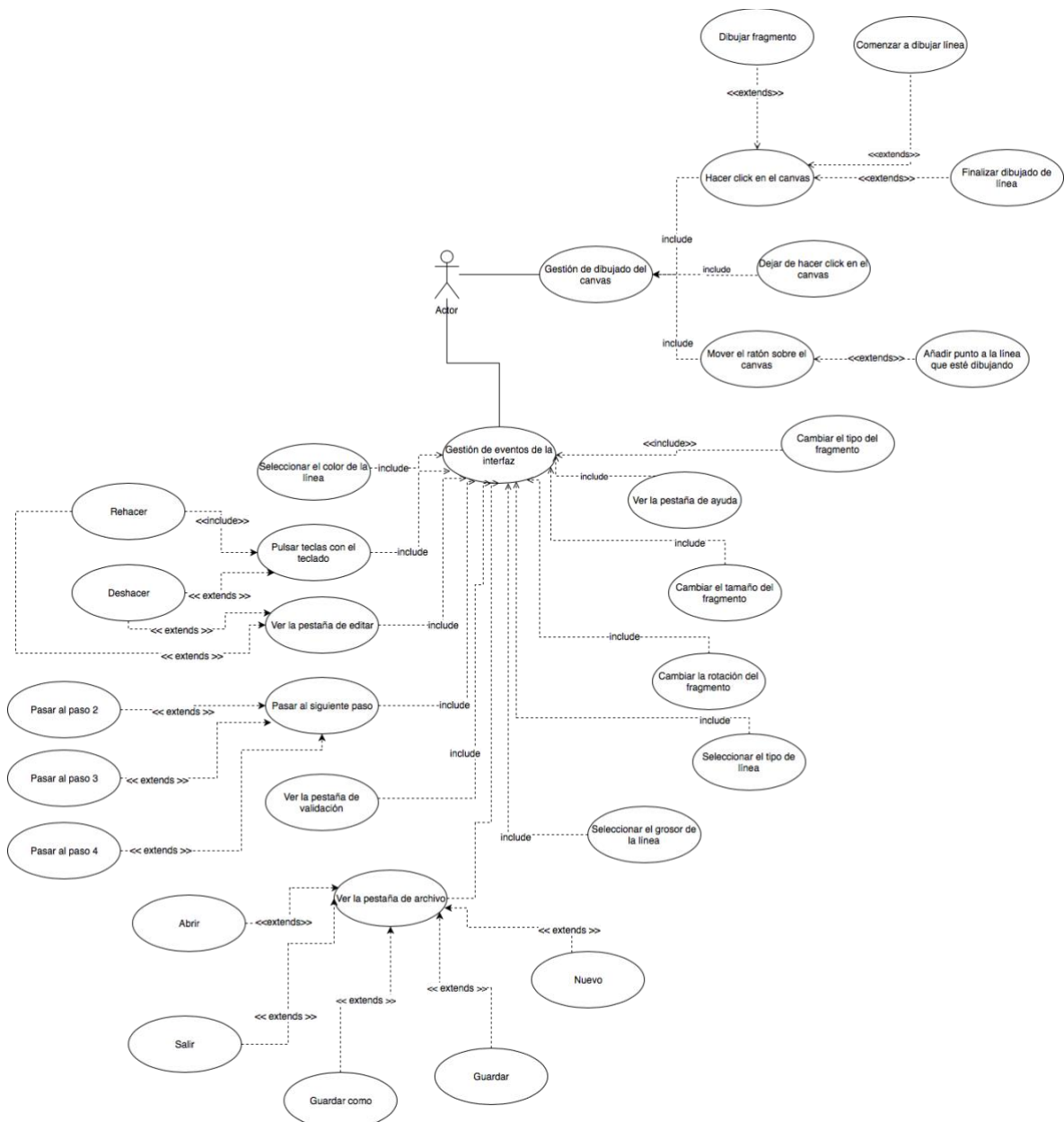


Ilustración 29. Diagrama de clases 6ª Iteración

- Diseño

En esta iteración los requisitos se corresponden únicamente a dar funcionalidad a los distintos elementos de la interfaz.

Dicha funcionalidad de cada uno de los elementos será aplicada desde el evento correspondiente a ese elemento mediante una llamada a la clase “Painter” que será la encargada de ejecutar dicha acción.

Para la exportación se usará el formato de exportación diseñado en iteraciones anteriores, asegurando la compatibilidad del sistema.

A continuación, se amplía el diagrama de clases de iteraciones anteriores para recoger todo lo necesario para la implementación de estos nuevos requisitos.

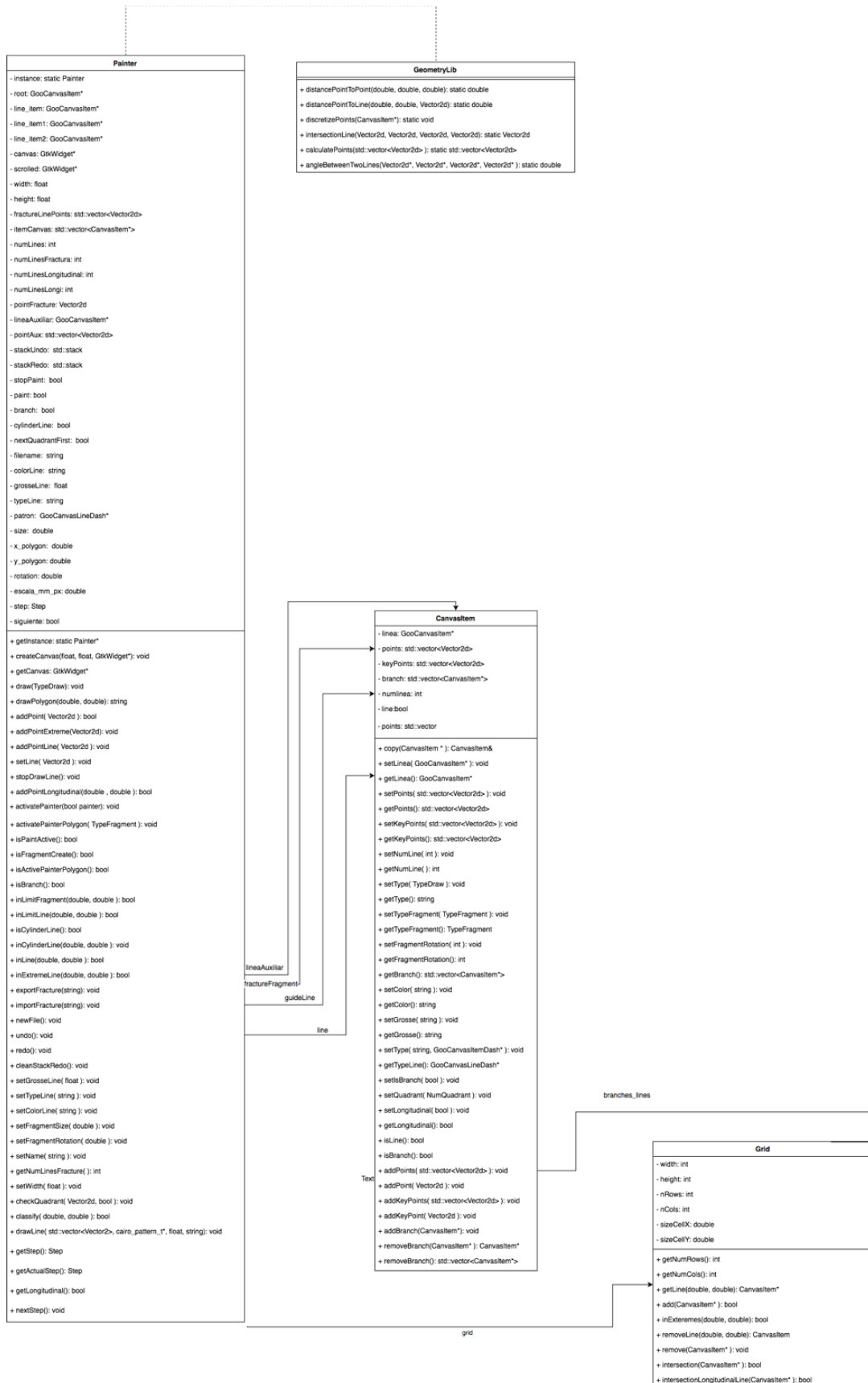


Ilustración 30. UML de iteración 6

- **Pruebas**

En esta fase se realiza un uso exhaustivo de la aplicación y se les pasa a distintos perfiles de usuarios en busca de posibles errores o mejoras.

3.3.3.7. 7ª Iteración. Implementación → Validador automático.

Esta iteración se corresponde con la implementación de un validador automático que, según la dirección de impacto, valorará el patrón de fractura producido siguiendo las estadísticas de diversos artículos científicos almacenadas en la herramienta.

- **Requisitos**

Requisitos funcionales:

- Deberá poder validar las características de un patrón en lo referente al tamaño de fragmento, número de líneas de fractura y longitud de las líneas de fractura según la dirección de impacto.
- Deberá poder validar el resultado de las líneas longitudinales presentes en el patrón según la dirección de impacto.
- Deberá poder validar que el número de líneas de fractura presentes en cada aspecto impactado es correcto o no según la dirección de impacto.

Requisitos no funcionales: en esta iteración no hay requisitos no funcionales.

A continuación, se muestra el diagrama de casos de uso actualizado a los nuevos requisitos.

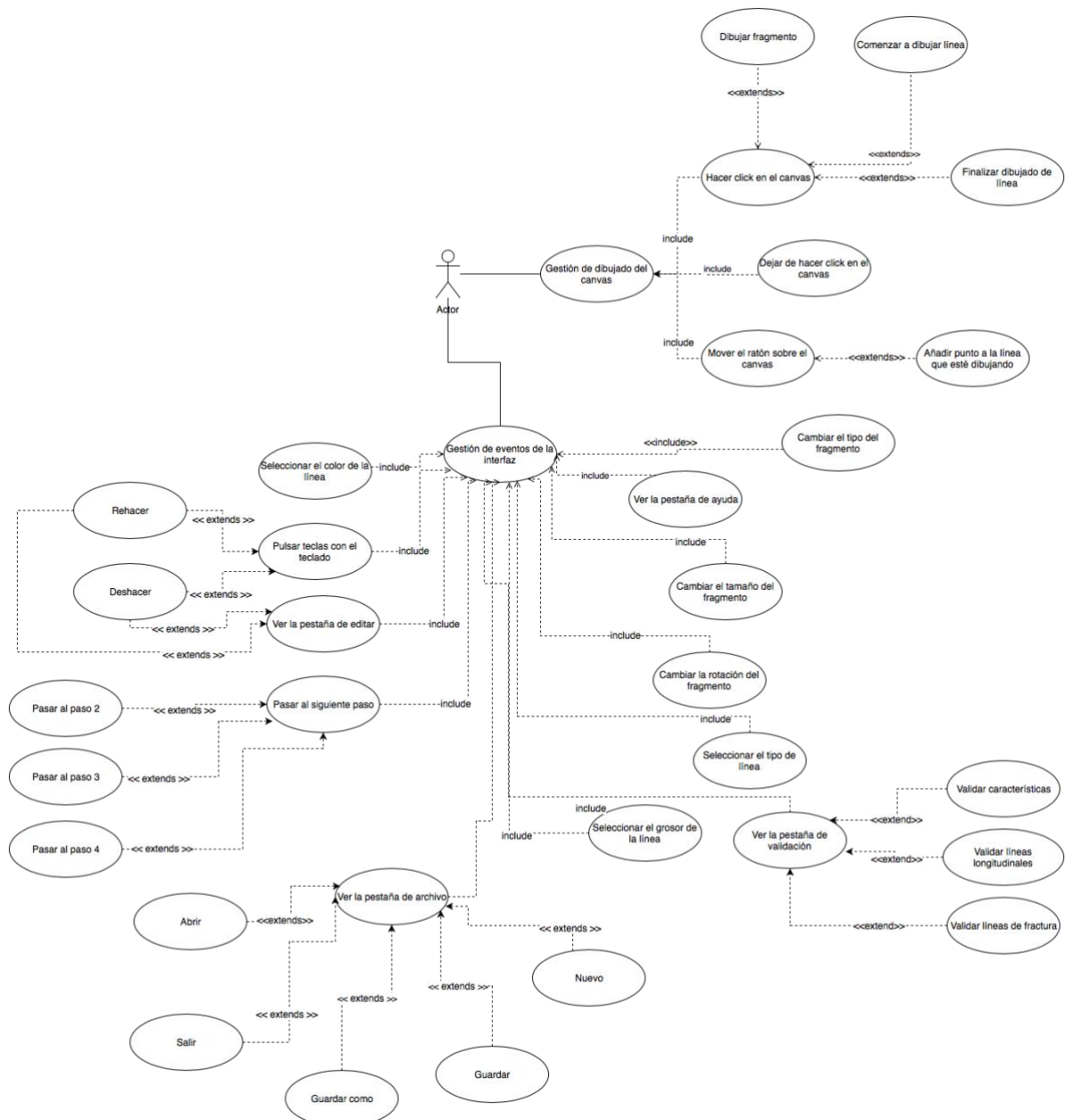


Ilustración 31. Diagrama de casos de uso final

- **Diseño**

En esta fase únicamente hay que añadir las funciones necesarias para realizar los distintos test estadísticos a nuestro patrón.

El diagrama de clases final sería el siguiente:

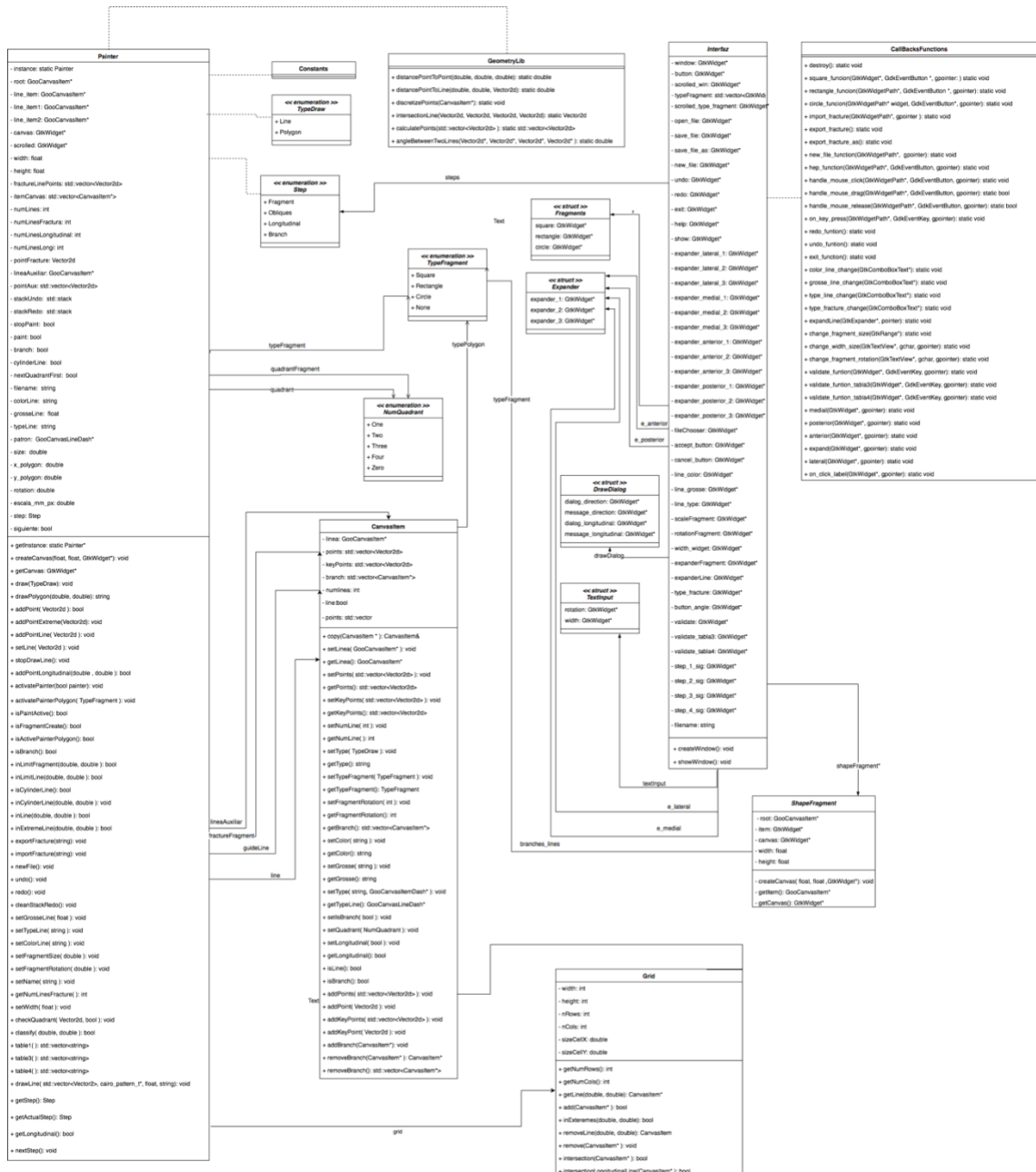


Ilustración 32. UML final

- **Pruebas**

Para probar esta funcionalidad se crearán diversos tipos de fracturas y se comprobará si se realizan de forma correcta los distintos test.

3.3.3.8. 8ª Iteración. Pruebas de patrones de fractura.

Esta iteración se va a centrar en la realización de distintos patrones de fractura para comprobar el correcto funcionamiento de la herramienta, pero, sobre todo, para tener patrones de fractura para validar posteriormente en otras iteraciones.

- **Requisitos**

En esta fase no se han definido requisitos.

- **Diseño**

En esta iteración no se ha definido fase de diseño.

- **Pruebas**

Esta iteración se va a centrar únicamente en la prueba de la herramienta mediante la realización de distintos patrones de fractura.

3.3.3.9. 9ª Iteración. Validación de resultados

En esta iteración no va a existir parte de requisitos ni de diseño, sino que esta iteración está orientada a la validación de la herramienta mediante test de [usabilidad](#), test de [expertos](#), validación automática y [validación](#) por parte de expertos de distintos patrones de fractura.

Las pruebas a su vez de dos pruebas: pruebas de usabilidad por parte de usuarios de todo tipo para ver si la elección de los elementos de la interfaz ha sido acertada y por otro lado las pruebas de expertos, en la que los expertos usarán la herramienta y validarán si los resultados de dicha herramienta podrían ser válidos, así como si la herramienta en sí es válida para tal cometido.

Finalmente, los expertos serán sometidos a varias pruebas en las que se les darán un conjunto de imágenes y deberán ser capaces de identificar cuales son imágenes correctas o no. Estos datos serán analizados y contrastados con los resultados dados por el validador automáticos de la herramienta para así vislumbrar si realmente dicho validador nos aproxima a un patrón de fractura correcto.

3.3.3.10. 10ª Iteración. Generación de patrones de fractura

Esta iteración al igual que la anterior no posee fase de requisitos, diseño ni pruebas. Está orientada únicamente a la generación de patrones de fractura para probar en otros sistemas y adjuntar con el proyecto a su finalización.

3.3.4. Caminos de acción

Una vez creada la herramienta, es el momento de la generación de los primeros patrones de fractura.

En la creación de un patrón de fractura, en caso de ser un usuario principiante con la herramienta, simplemente hay que seguir los pasos indicativos para llegar al diseño de fractura deseado. En caso de usuario experimentado, simplemente ya conocerá los pasos a seguir y sólo tendrá que preocuparse de hacerlos.

3.3.4.1. Creación de un patrón

Para realizar la generación de un patrón de fractura hay que tener en cuenta una serie de factores y conocimientos para que podamos realizarla de forma adecuada.

- **Elección de fractura con desprendimiento de viruta o sin él**

- a) Si quiero una fractura con desprendimiento de viruta, simplemente se selecciona la forma del fragmento desprendido y se sitúa en el lienzo de dibujado.

- b) Si quiero una fractura sin desprendimiento de viruta, simplemente se hace clic sobre la posición del lienzo de dibujado donde se quiera comenzar a pintar las líneas.

- **Dibujado de líneas oblicuas**

Se dibujan las líneas oblicuas a partir del fragmento, pudiendo dar una vuelta completa al hueso o parando antes mediante el pulsado de la tecla “esc”.

- **Dibujado de líneas longitudinales**

- a) Si se quiere dibujar líneas longitudinales, se realiza o bien clicando en cualquier parte del aspecto no impactado que intersecte con otra línea o en el aspecto impactado si intersecta con el fragmento o punto de fractura inicial.
- b) Si no se quiere dibujar líneas longitudinales, se omite el paso clicando en la flecha correspondiente al paso actual.

- **Dibujado de ramificaciones**

- a) Si se quieren dibujar ramificaciones, se hace clicando sobre una línea ya dibujada que no sea longitudinal.
- b) En caso contrario, no se realiza.

Opcionalmente se podrá validar la fractura para obtener una estimación de su corrección, este aspecto viene explicado más detalladamente en apartados siguientes, o bien guardar la fractura y exportarla o salir.

3.3.4.2. Ejemplo de generación de un patrón

En el siguiente ejemplo se recoge la creación de un patrón de fractura con fragmento de viruta paso a paso.

- **Abrimos la aplicación y aparece la siguiente interfaz.**

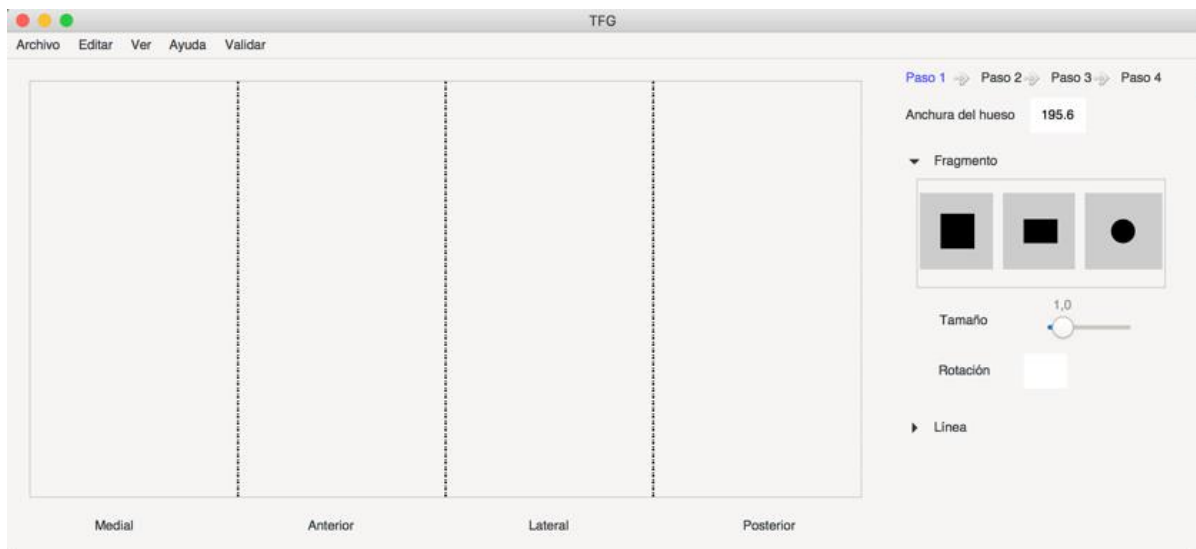


Ilustración 33. Aspecto de la aplicación inicial

- **Establecer el fragmento desprendido**



Ilustración 34. Selección de la forma del fragmento desprendido

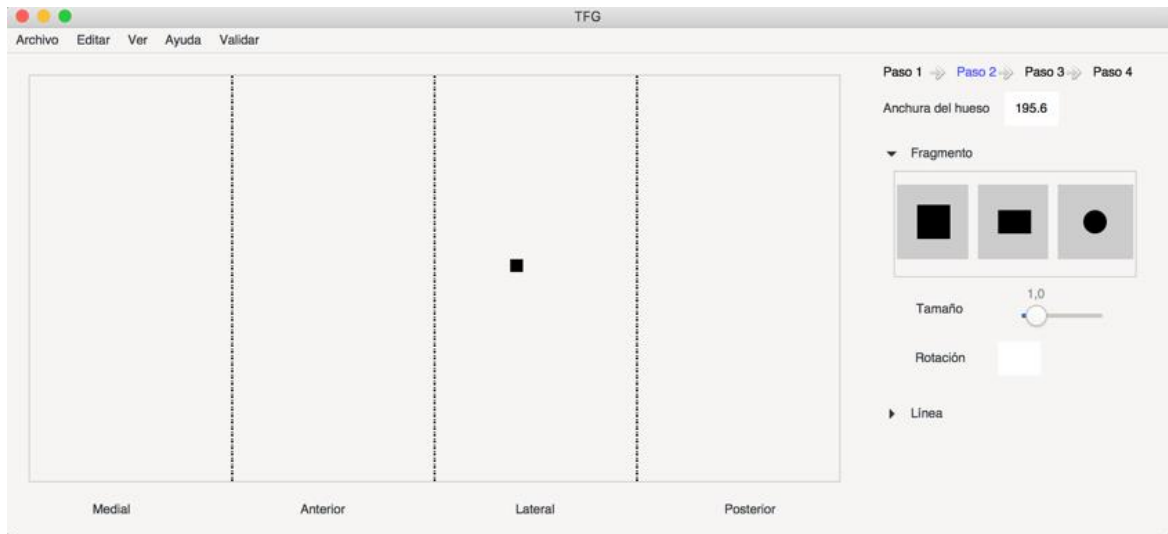


Ilustración 35. Dibujado del fragmento desprendido

- **Dibujar las líneas oblicuas**

Se clic en la posición donde se quiere empezar (dentro del fragmento o en sus límites) y se comienza a dibujar. Al llegar a una línea de separación del canvas se hace clic en el aspecto siguiente para que la línea se continúe dibujando por él.

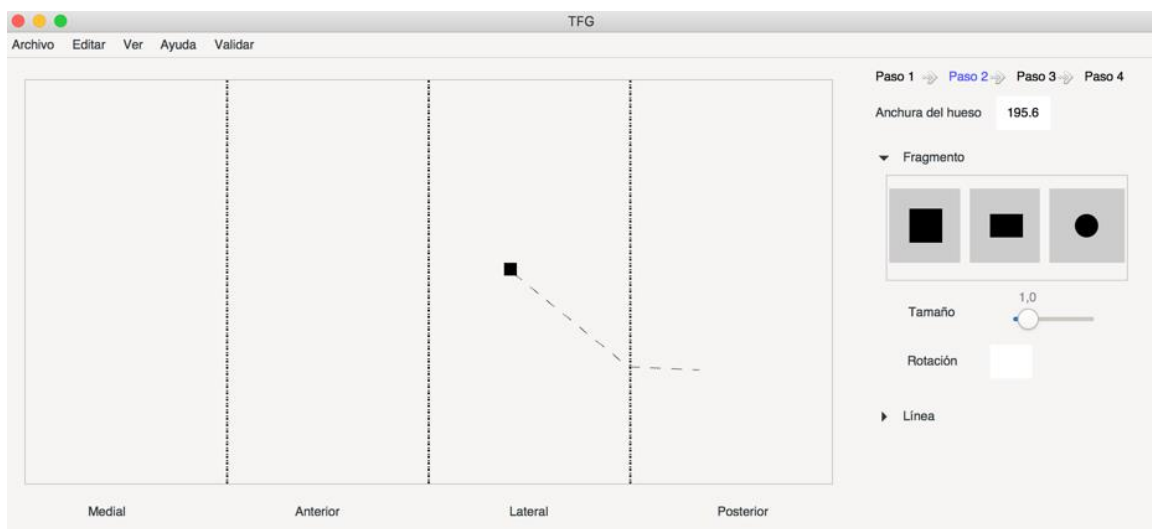


Ilustración 36. Dibujado de líneas oblicuas

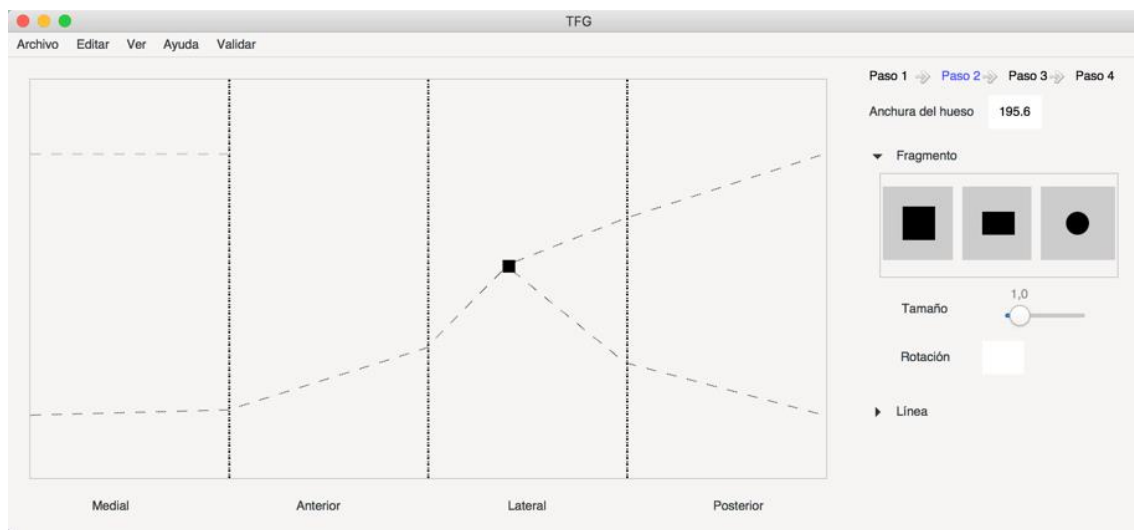


Ilustración 37. Dibujado de líneas oblicuas con línea guía

■ Líneas longitudinales

Ahora paso al dibujado de líneas longitudinales, para lo cuál dibujo una en el aspecto impactado y otra en el que se encuentra entre el contra lateral y el impactado.

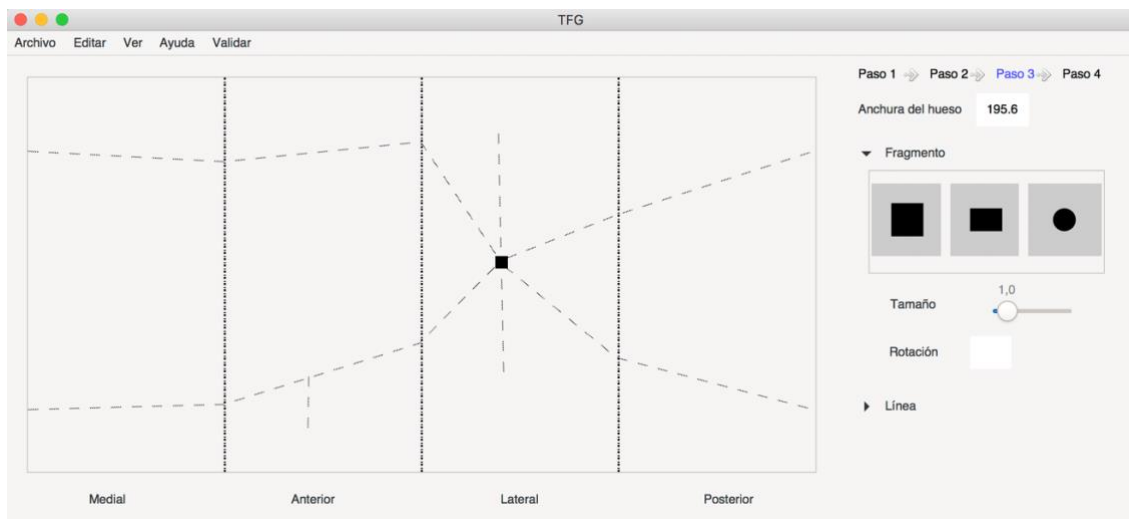


Ilustración 38. Dibujado de líneas longitudinales

Para pasar al siguiente paso, pulso la flecha del paso 3 (situado arriba a la derecha en la interfaz).

- **Dibujado de ramificaciones o líneas secundarias.**

Como se puede observar en la imagen, se puede ver que ya ha pasado al paso 4.

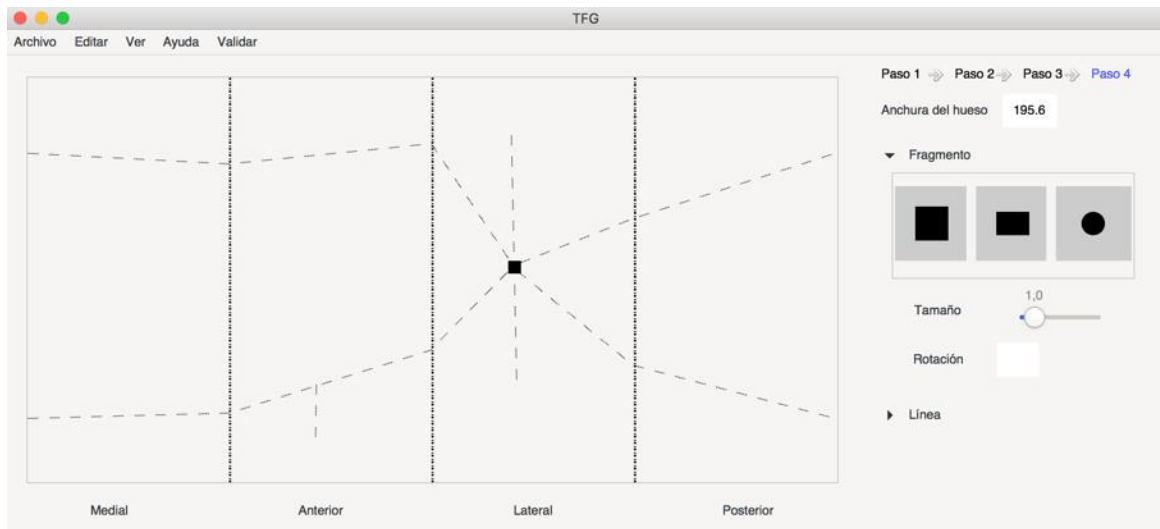


Ilustración 39. Paso 4 de dibujado

Para dibujar este tipo de líneas pulso con el ratón en una línea ya dibujada y continúo dibujando a partir de la misma.

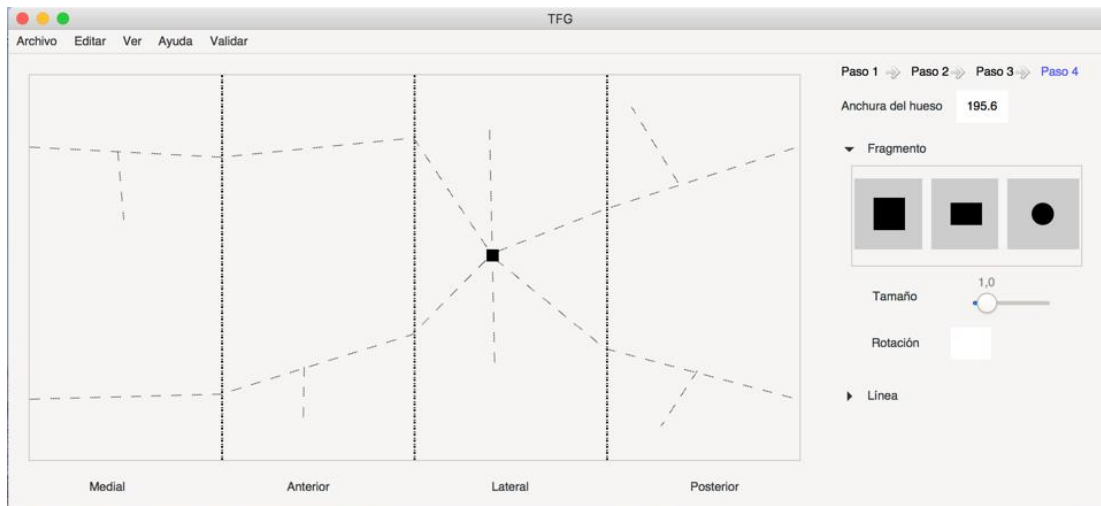


Ilustración 40. Dibujado de líneas secundarias

■ Guardamos el patrón de fractura

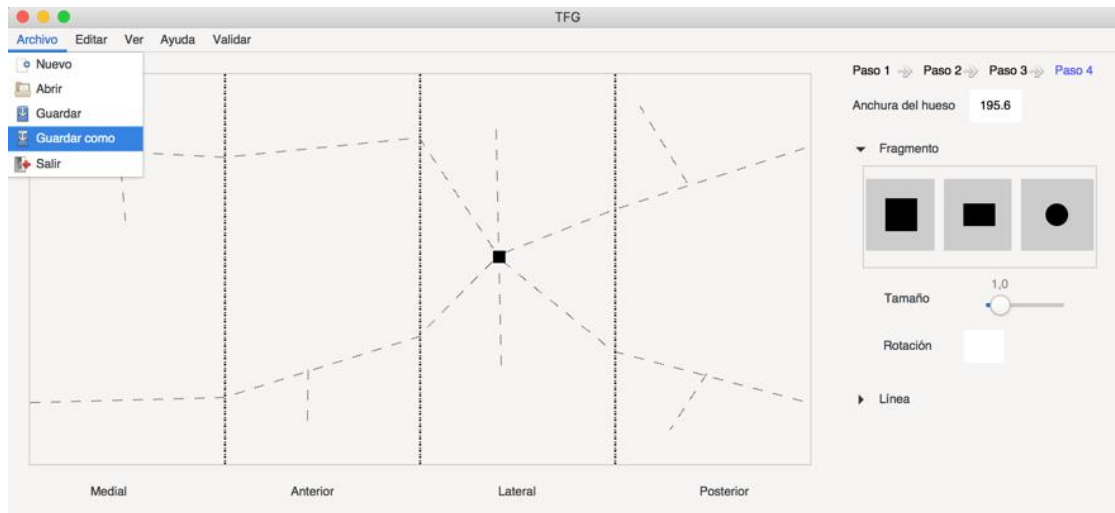


Ilustración 41. Guardado del patrón de fractura

El resultado del guardado es el siguiente fichero:

```
#Si la línea empieza en # es un comentario
#Cada dato se separa por un espacio en blanco, no tabulaciones
#Cada línea define un objeto geométrico diferente
#Si la línea empieza en $ el objeto de esa línea es parte del chip fragment
#El formato es el siguiente
#Tipo Nmero-de-puntos Punto-1-X Punto-1-Y Punto-2-X Punto-2-Y ... (Opcional)P Color-r
Color-g Color-b Opacidad Grosor-línea Codigo-línea (Opcional) B Color-r Color-g Color-b
Opacidad
#Tipo, puede ser 0 - Polígono o 1 - Polilínea, 2 Square, 3 Rectangle, 4 Hexagon, 5 Circle, 6
Sin impacto
#La parte opcional P se refiere a la herramienta lápiz, y la parte B a la herramienta
brocha(relleno)
#El relleno(brocha) solo se aplica a polígonos
#Los colores y opacidad son en base 255
#Las opciones para código de línea son 1 - Línea sólida, 2 - Línea rayada, 3 - Línea punteada, 4
- Línea(Raya Punto), 5 - Línea(Raya Raya Punto)
$ 0 452 207 452 197 462 197 462 207 P 0 0 0 255 2 2 B 0 0 0 255

1 460 198 568 110 758 60 P 0 0 0 255 2 1
0 60 189 65 378 124 453 200 P 0 0 0 255 2 1
0 60 189 65 378 124 453 200 P 0 0 0 255 2 1
1 462 206 569 248 758 309 P 0 0 0 255 2 1
0 309 189 299 379 318 456 203 P 0 0 0 255 2 1
0 309 189 299 379 318 456 203 P 0 0 0 255 2 1
1 454 326 459 95 P 0 0 0 255 2 1
1 271 92 270 33 P 0 0 0 255 2 1
1 640 274 594 349 P 0 0 0 255 2 1
1 659 88 623 33 P 0 0 0 255 2 1
1 87 305 93 237 P 0 0 0 255 2 1
```

Ilustración 42. Patrón de fractura generado importado

3.3.5. Decisiones tomadas en la implementación.

En cuanto a la implementación, es la fase del proyecto que ha tenido mayor duración y hay algunas decisiones tomadas que merecen ser justificadas.

3.3.5.1. Uso de grid.

En primer lugar, para poder dibujar líneas en el canvas, necesitaba hacer clic en el lugar donde quería que comenzasen a dibujarse dichas líneas.

Uno de los aspectos clave en el hecho de poder hacer clic en el canvas de dibujado, es que se puedan tanto dibujar líneas de cero como a partir de fragmentos o de otras líneas de fractura. Para que el dibujado se haga de forma eficiente, el comprobado de si existe una línea en la posición en la que se ha hecho clic, existe un fragmento o no hay nada he decidido realizarlo con la ayuda de un grid para agilizar el proceso de comprobación de si hay algún elemento en dicha posición de ratón en el cual se almacenan los distintos elementos dibujados. De esta forma, cada vez que dibujo una línea nueva, se almacena un puntero a dicha línea en las casillas del grid que contienen a esa línea, junto con el resto de elementos dibujados, con una previa comprobación de si existe o no un elemento en la posición. En este proceso de comprobación, se lograría saber si existe un elemento en la posición del clic en $O(1)$, ya que se accedería a la posición del grid que debe contener dicho punto en una sola operación y tras esto solo habría que comprobar si efectivamente existe la línea o algún elemento ahí, que sería $O(n)$, pero teniendo en cuenta que dicha casilla apenas contendrá dos o tres líneas, en lugar de $O(n)$, que sería la eficiencia en caso de hacer una búsqueda lineal por todos los elementos dibujados. Por tanto, aunque a priori parezca que ambos trabajan con una eficiencia de $O(n)$ en teoría, en la práctica es mucho más eficiente al no tener que recorrer todos los elementos dibujados, sino dos o tres como mucho.

3.3.5.2. Dibujado de líneas.

En cuanto al dibujado de las líneas, se le ha dado soporte tanto a líneas oblicuas, transversales y longitudinales como a ramificaciones, así como a posibles ramificaciones de ramificaciones. Esto se ha realizado encapsulando cada línea en una clase “CanvasItem”, que mantenga todas las características de una línea, incluido punteros a sus ramificaciones hijas.

Centrándome en el tema del dibujado de las líneas, he encontrado varios tipos de líneas según los artículos científicos, como ya he mencionado anteriormente, cada uno de los cuales tienen sus características, haciendo que en algunos casos el dibujado de la línea sea distinto. Este hecho se ve claramente en la diferencia de dibujado entre las líneas transversales y longitudinales con respecto a las oblicuas. Para el dibujado de las líneas oblicuas es estrictamente necesario controlar que dicha línea se dibuje en trazo, es decir, que no se deje la línea a medio dibujar, se dibuje otra y luego se vuelva a la anterior, ya que las líneas oblicuas, por definición en esta área, si llegan al extremo del canvas, significa que la línea recorre el hueso de forma horizontal como si de un cilindro se tratase. Por ello, he solventado este problema surgido por una restricción estricta derivada de la creación de una fractura, mediante el dibujado de una línea provisional al otro lado del canvas, por donde dicha línea de fractura debe continuar, exactamente a la altura correcta, para facilitar al usuario su continuación sin que tenga que buscar el punto en la altura correcta con el ratón para poder continuar el dibujado. Simplemente con pinchar en el cuadrante correcto, en el cual está la línea auxiliar o línea guía, la línea guía desaparecerá y se comenzará a dibujar una línea desde el punto correcto hasta donde el usuario lleve dicha línea.

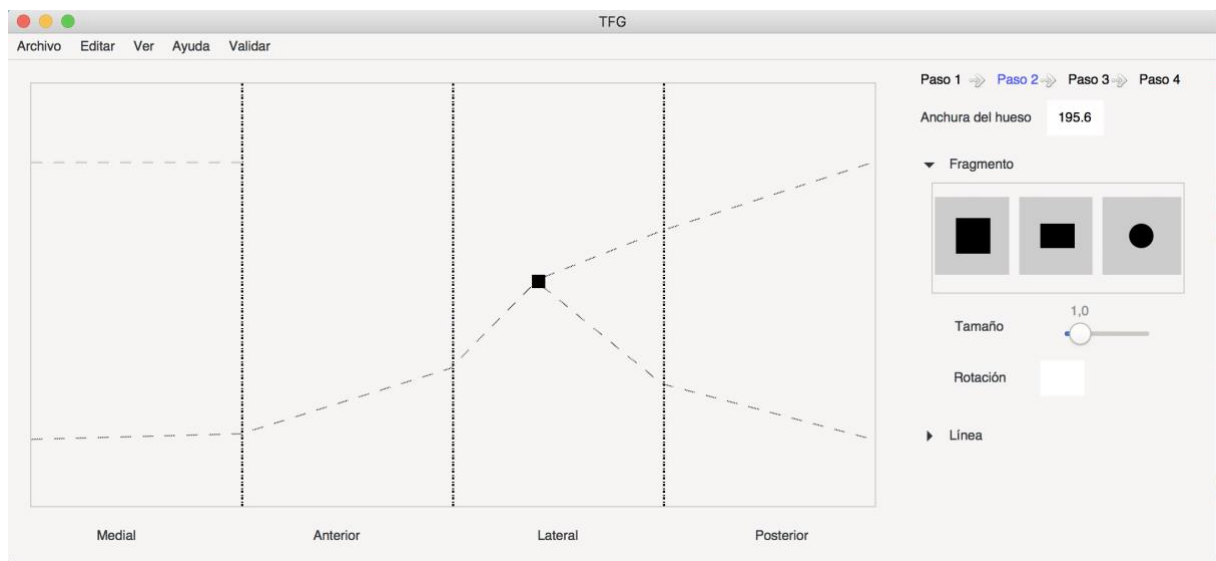


Ilustración 43. Línea oblicua con línea guía en el primer cuadrante

Además, en el dibujado de dichas líneas, he creído necesario, que para pasar de un cuadrante a otro sea necesario hacer clic en el cuadrante al que se quiera pasar, cerca de la línea separadora, para adquirir un mayor control del dibujado de las líneas y acumular menor error de dibujado. Para ello, se hace una comprobación de cuadrante comparando con la posición de las distintas líneas verticales punteadas del canvas, obteniendo así si la línea debe continuar por el cuadrante actual o si deberá pasar al cuadrante siguiente.

En el caso del dibujado del resto de tipos de líneas, no hay que tener en cuenta nada de esto, solamente el punto de inicio y el punto de fin y, en el caso de querer hacer ramificaciones o continuar alguna línea, sólo habría que comprobar a qué línea pertenece y añadir la nueva línea o bien como ramificación de la primera o modificar la primera para continuarla con los nuevos datos.

3.3.5.3. Dibujado del fragmento.

Continuando con el dibujado de líneas, he tenido que diferenciar entre el dibujado de líneas y el dibujado de polígonos, ya que he considerado que la mejor opción para el dibujado del fragmento desprendido es mediante la elección de una de las formas de fragmentos predefinidos por el programa, aunque esto más adelante podría cambiar. Por el momento, existen tres tipos de fragmentos

que serían el cuadrado, el rectángulo y el círculo, aunque son soportados otros tipos de formas de fragmento si se trata de una fractura importada, en cuyo caso se mantiene la forma del fragmento de forma fidedigna. A dichos fragmentos, para el cálculo del punto a partir del cual se considera que se ha seleccionado el fragmento para dibujar líneas de fractura o para girarlo o escalarlo, se ha calculado el centro de los distintos polígonos que se obtengan mediante distintas fórmulas geométricas. Para los polígonos que no sean una circunferencia, su centro calcula mediante la intersección de sus diagonales, siendo el punto donde intersectan el centro del polígono. Para el caso de la circunferencia, simplemente ya sabemos el centro para el dibujado de la misma, ya que es el punto que se selecciona para ponerla en el canvas de dibujado. Otro dato relevante en el dibujado de un fragmento es que se guarda el cuadrante en el cual se ha depositado el mismo y dicha posición da nombre al tipo de fractura, pudiendo ser medial, anterior, posterior o lateral.

3.3.5.4. Restricciones.

Por otro lado, al tener el requisito fundamental de que las líneas tienen que ser realmente polilíneas que no se corten consigo mismas y que no se enreden entre sí como si de un nudo se tratase, al igual que sólo contenga los puntos que realmente forman parte de la línea, ya que mientras que la línea se va moviendo por el canvas sin fijarla, va almacenando todas las posiciones por las que pasa la línea, generando un problema grave, ya que cuando se hacía clic en cualquier parte del canvas por la que había pasado la línea, aunque realmente no estuviera, consideraba que existía dicha línea ahí. Para ello, he tenido que implementar una función que discretiza los puntos existentes entre dos puntos clave, que son los dos puntos donde se haría clic para dibujar la línea. Esta función básicamente lo que hace es quedarse con los puntos que realmente están sobre la recta que une dos puntos clave, consiguiendo así tener una línea con los puntos correctos.

Otro de los problemas a resolver mediante geometría, ha sido comprobar que un punto donde se hace clic para establecer el inicio de una línea está realmente sobre una línea. Para ello, he implementado un método que calcula la

distancia de un punto a la línea creada por dos puntos. Esto lo uso para calcular la distancia del punto a las líneas que hay en la casilla del grid correspondiente a esa posición del punto y así encontrar si realmente se ha pinchado sobre una línea o fuera de esta.

$$t_0 = \frac{\vec{d} \cdot (\vec{A} - \vec{P})}{\vec{d} \cdot \vec{d}}, D = \begin{cases} |\vec{A} - \vec{P}|, & t_0 \leq 0 \\ |\vec{A} - (\vec{P} + t_0 \vec{d})|, & 0 \leq t_0 < 1 \\ |\vec{A} - (\vec{P} + \vec{d})|, & t_0 \geq 1 \end{cases}$$

Ecuación 1. Distancia de un punto a una recta

Por motivos similares, también se ha calculado la distancia entre dos puntos, en este caso, la distancia entre dos puntos lo uso, junto con otros cálculos, para detectar si se ha pinchado sobre un fragmento circular o se ha pinchado fuera del mismo.

$$d(AB) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Ecuación 2. Distancia entre dos puntos

Otro método geométrico implementado es la intersección entre dos líneas. En este caso, se usa para evitar que dos líneas que no sean ramificaciones intersecten entre sí.

$$(P_x, P_y) = \left(\frac{(x_1 y_2 - y_1 x_2)(x_3 - x_4) - (x_1 - x_2)(x_3 y_4 - y_3 x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}, \frac{(x_1 y_2 - y_1 x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 y_4 - y_3 x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \right)$$

Ecuación 3. Punto de intersección entre dos líneas

3.3.5.5. Importación/exportación de patrones.

En lo referente a temas de exportación e importación, al realizar la exportación e importación de las líneas, solo exporto los puntos clave de cada polilínea, ya que se acordó así con el grupo de investigación, que son los puntos en los que cambia la curvatura de cada línea. Esto genera un problema, que ya vi al importar a mi sistema fracturas de otros sistemas, que es que cuando se importa una fractura y la cargo en el canvas de dibujado, si se quiere editar, no se tienen los puntos de la línea para poder buscar en el grid si en el punto donde se quiere dibujar ya hay algún otro elemento. Para ello, he creado un método que me restaura los puntos que hay entre dos puntos clave al realizar la importación, para que así pueda detectar que está la línea en todas las posiciones del grid por las que pase dicha línea. El número de puntos que calculo por cada segmento de línea depende de la longitud de esa línea, para que tenga un número de puntos homogéneo y que a la par no sea excesivo.

3.3.5.6. Pasos de dibujado.

Se ha creado una secuencia de pasos que va de 1 a 4 y que se corresponden a las distintas fases de dibujado.

La primera fase se corresponde con el dibujado del fragmento de viruta o punto de fractura y pasa de forma automática al hacerlo, pudiendo ser opcional tener fragmento y derivar en tener punto de fractura o viceversa.

La segunda fase es la fase de dibujado de líneas oblicuas y pasa de forma automática cuando se ha completado su dibujado. Se corresponde a la fase de dibujado de líneas que salen del fragmento y se extienden con alguna inclinación de forma horizontal al canvas de dibujado.

En la fase 3 se dibujan líneas longitudinales, en cuyo caso podemos elegir entre dibujar o pasar de forma manual cuando queramos pasar a la siguiente fase. Este tipo de líneas se corresponden con líneas totalmente verticales.

En la última fase se realizan las ramificaciones o líneas secundarias.

3.3.5.7. Deshacer/rehacer operaciones

La funcionalidad de deshacer operaciones, ha sido más problemática de lo esperado, debido a la estructura interna de los elementos de `goo_canvas`.

Para su implementación, he creado una pila de operaciones en la que cada vez que inserto una nueva línea, inserto un vector con las referencias a dichas líneas, de forma que cuando quiero deshacer una operación tan solo tengo que restaurar el canvas al estado anterior de la aplicación, situado en la pila de operaciones. Tras deshacer dicha operación guardo la lista de elementos que acabo de deshacer en otra pila, que se corresponde a la pila de rehacer operaciones, que funciona de igual forma que la de deshacer. Además, en la fase de deshacer se tiene en cuenta la fase en la que se encuentra y si se ha desecho por ejemplo la fase de dibujado de líneas longitudinales, pues vuelve a su fase correspondiente de líneas oblicuas.

3.3.5.8. Validación.

Para la validación de expertos, he implementado varias funciones que hacen recuento de distintos tipos de líneas. Para ello, he necesitado distinguir cuando una línea es longitudinal y cuando no. He considerado que una línea es longitudinal si una línea tiene pendiente infinita, por lo que me he basado en esa ecuación matemática para calcularlo. Para el cálculo del resto de recuentos de líneas, simplemente se ha hecho comprobando el tipo de línea y que esté en el cuadrante correspondiente, junto con los datos estadísticos de la documentación seguida.

3.4. Exportación e integración en otros sistemas

Para realizar el formato de exportación he tenido en cuenta los distintos parámetros necesarios de cada elemento dibujado en el canvas en el momento de realizar la exportación o guardado que necesitaría para poder recrear el patrón si quisiera volver a introducirlo en mi sistema y visualizarlo.

Se ha decidido que en el fichero pueda haber comentarios explicando la estructura del fichero. Para indicar que se trata de un comentario se pondrá un '#' al inicio de la línea y ya añadirá el resto de texto con total normalidad.

Por otro lado, para diferenciar entre fragmento o línea, usaremos el símbolo '\$' al inicio de la línea para indicar que es un fragmento, en otro caso no aparecerá dicho símbolo, sino que directamente aparecerá el tipo de línea.

A continuación de si es línea o fragmento, se añadirán todos los puntos clave del elemento a dibujar. Y tras esto, se podrán incluir una serie de parámetros considerados opcionales como son:

- P 0 0 0 255 2 1, hace referencia a la herramienta de lápiz, expresada en RGBA, el grosor de la línea y el tipo de línea (1 → sólido, 2 → rayada, 3 → punteada, 4 → línea raya punto, 5 → línea raya raya punto).
- B 0 0 0 255, hace referencia a la herramienta de brocha expresada en RGBA, esta herramienta solo es aplicable en polígonos.

```
#Si la linea empieza en # es un comentario
#Cada dato se separa por un espacio en blanco, no tabulaciones
#Cada linea define un objeto geométrico diferente
#Si la linea empieza en $ el objeto de esa línea es parte del chip fragmento
#El formato es el siguiente
#Tipo Número-de-puntos Punto-1-X Punto-1-Y Punto-2-X Punto-2-Y ... (Opcional)P Color-r
Color-g Color-b Opacidad Grosor-línea código-línea (Opcional) B Color-r Color-g Color-b
Opacidad
#Tipo, puede ser 0 - Polígono o 1 - Polilínea
#La parte opcional P se refiere a la herramienta lápiz, y la parte B a la herramienta
brocha(relleno)
#El relleno(brocha) solo se aplica a polígonos
#Los colores y opacidad son en base 255
#Las opciones para código de línea son 1 - Línea sólida, 2 - Línea rayada, 3 - Línea
punteada, 4 - Línea(Raya Punto), 5 - Línea(Raya Raya Punto)
$ 0 255.679 190.193 295.162 215.839 314.287 190.54 276.979 163.399 P 0 0 0 255 2 1 B
0.501961 0.501961 0.501961 255
1 669.216 52.9275 654.38 203.344 650.719 324.244 P 0 0 0 255 2 2
1 300.928 208.211 380 298.618 475 353.809 651.003 314.875 P 0 0 0 255 2 2
1 273.974 202.077 190 294.052 95 350.763 -0 332.995 P 0 0 0 255 2 2
1 296.871 177.87 380 92.9205 475 37.8846 668.909 56.0395 P 0 0 0 255 2 2
1 269.002 173.434 190 91.6232 95 36.8791 -0 46.6874 P 0 0 0 255 2 2
1 283.369 208.179 279.487 285 P 0 0 0 255 2 2
1 21.57 44.4604 40.7371 251.022 P 0 0 0 255 2 2
1 122.474 334.362 136.534 361.082 P 0 0 0 255 2 2
1 324.969 149.156 363.121 168.044 P 0 0 0 255 2 2
1 592.676 327.777 604.164 354.283 P 0 0 0 255 2 2
1 760 332.995 651.071 312.621 P 0 0 0 255 2 2
1 760 46.6874 668.904 56.0927 P 0 0 0 255 2 2
```

Ilustración 44. Ejemplo de fichero de exportación

En cuanto a la extensión del fichero de exportación, he decido realizar la exportación con la extensión pf2d, ya que de esta forma serían compatible los patrones generados con mi herramienta con otro software que me consta que usa dicho formato en el departamento.

Por otro lado, para la importación puede leer otro tipo de extensiones además de “pf2d”, como por ejemplo “.txt”.

Una vez realizado el exportador, se ha probado a realizar la exportación del fichero y el cargado del mismo en otros sistemas con total compatibilidad, al igual que se ha probado a importar de otros sistemas con igual éxito.

4. Validación de los patrones de fractura y usabilidad

En este apartado se va a proceder a analizar los resultados dados por los distintos test y los resultados dados del análisis de patrones de fractura creados en la herramienta por expertos.

4.1. Pruebas de usabilidad

En la busca de un producto intuitivo y fácil de usar, mientras se iba realizando la herramienta, esta ha sido ofrecida a distintos tipos de usuarios, con distintos conocimientos informáticos y franjas de edad para detectar los aspectos menos entendibles de la aplicación y poder modificarlos hasta lograr una herramienta lo más intuitiva posible.

Para el control de estos aspectos, se ha entregado la herramienta y se ha dejado al usuario usarla sin explicarle absolutamente nada, simplemente dejándole una imagen de un patrón de fractura, ya que los usuarios podían ser o no expertos en la materia, y diciéndoles que intentaran dibujar algo similar. Tras su uso, se les entrega un cuestionario (Anexo II) y se analizan los resultados.

- Análisis de los resultados

En la siguiente tabla se recogerán los datos obtenidos en una muestra de 10 encuestas.

La correspondencia de las abreviaturas de la tabla es:

U = Universidad, B = Bachiller, E = ESO, T = Técnica, S = Superior

Pregunta	1	2	3	4	5	6	7	8	9	10	Media
Sexo	F	F	M	M	M	M	F	M	F	F	
Edad	26	52	55	22	28	40	45	39	18	27	35,2
Nivel académico	U	B	E	U	T	U	S	T	B	U	
1. Encuentro la herramienta fácil de usar	5	4	3	4	5	4	3	4	4	4	
2. Requería de ayuda para realizar las tareas solicitadas	4	4	3	4	4	4	4	4	4	3	3.8
2. Considero al producto fácil de usar para cualquier persona con conocimientos en el área y con capacidad de manejar una computadora o sistema	5	5	5	5	5	5	5	5	5	5	5
3. Considero que no se requiere conocimiento previo para realizar las tareas solicitadas	3	4	2	3	5	3	3	4	5	4	3.6
4. Mensajes e instrucciones ambiguos o incorrectos	5	5	5	5	5	5	5	5	5	5	5
5. Número de pasos excesivos para realizar una tarea	5	5	5	5	5	5	5	5	5	5	5
6. Considero fiable al producto	5	5	5	5	5	5	5	5	5	5	5
7. Considero seguro al producto software	5	5	5	5	5	5	5	5	5	5	5
9. Cometí uno/muchos error/errores durante la realización de las tareas, a causa de no leer claramente un mensaje o indicación del producto software	5	3	5	5	5	4	5	4	5	3	4.4
10. Cometí uno/muchos error/errores durante la realización de las tareas a causa de que algún/os mensaje/s o indicación/es de la herramienta software no estaban claros	5	5	2	5	5	5	3	5	5	5	4.5
11. El producto software tiene mensajes o instrucciones que no son claros para poder llevar a cabo una tarea	5	5	5	5	5	5	5	5	5	5	5
12. Considero elegante o atractivo al producto software (combinación de colores adecuada, tamaño de letra legible, etc.)	5	5	5	5	5	5	5	5	5	5	5
13. Estoy cómodo con el software	5	5	5	5	5	5	5	5	5	5	5

Tabla 6. Resultados de test de usabilidad

De los resultados obtenidos de esta muestra de usuarios, podemos ver que es una muestra muy variada, de distintas edades y niveles de estudios. Los resultados que se ven claramente a simple vista es que la interfaz no presenta errores en sus mensajes o mientras se realizan las tareas, así como se ve que tiene una interfaz clara y que hace sentir cómodo al usuario con ella.

En lo referente a la facilidad de uso de la herramienta, vemos claramente que la herramienta en sí no es demasiado complicada de usar, ya que solamente está un punto por debajo de la facilidad completa de uso de la herramienta. De igual forma, debido a esto, la misma proporción aproximadamente necesita ayuda, ya que titubea en cierto grado, aunque sea bajo, en el uso de la herramienta.

Los usuarios coinciden en que para el uso de la herramienta se requieren conocimientos ya sean en el área de estudio o en informática, por lo que no supondría ningún problema para nuestros usuarios objetivo ya que serán expertos en el área.

Por otro lado, se denota que algunos usuarios no leen las indicaciones en la interfaz, por lo que se producen errores de tipo, aunque en baja proporción.

4.2. Validación automática

Las fracturas creadas por los distintos usuarios pueden ser más o menos realistas dependiendo de su capacidad para el trazado de dichas líneas, así como de su conocimiento en cuanto a los distintos tipos de líneas de fractura se refiere, pudiendo hallarse patrones de fractura correctos, pero también incorrectos.

Para asegurarse un patrón de fractura correcto, es necesario tener otros patrones similares de referencia que puedan guiar al usuario en su ardua tarea de construcción de un patrón de fractura.

En el artículo de Forensic Science International[13], se hace referencia a los distintos tipo de fractura así como a las distintas características de las mismas, ofreciendo en uno de sus apartados diversas estadísticas dependiendo de factores como las características métricas del fragmento, de las líneas de fractura,... dependiendo de la dirección de impacto, la distribución de las líneas longitudinales y el número medio de líneas de fractura dependiendo de la dirección de impacto, etc.

En la búsqueda por verificar los distintos patrones de fractura, se ha conseguido obtener una comparativa cuantitativa en lo referido a las distintas características que definen un tipo de fractura concreto y el patrón de fractura creado.

En el artículo mencionado anteriormente, se hacen distintos tipos de test basándose en la dirección del impacto (impacto lateral, anterior, medial y posterior) y en la carga axial que pueda experimentar dicho impacto.

Para el análisis y verificación de nuestro patrón de fractura no se ha encontrado necesario hacer la comparativa con los test de carga axial debido a que con la herramienta únicamente se puede crear el patrón sin considerar parámetros externos que tengan influencia en dicha generación, ya que eso se encuentra fuera de objeto de estudio en este proyecto.

En cuanto al resto de test, se han encontrado varias tablas estadísticas que los aplican. En concreto tenemos:

- **Análisis de las características métricas de una fractura bajo distintas direcciones de impacto.**

Para la verificación de nuestro patrón son necesarios únicamente los parámetros referidos al número de líneas de fractura, la longitud de la línea de fractura o del polígono resultante y el tamaño del fragmento desprendido (todo ello en mm), ya que el resto de parámetros están fuera de nuestro estudio.

Measurements	Direction of impact								P value [*]
	Lateral impact Test 1 (T ₁)		Posterior impact Test 2 (T ₂)		Medial impact Test 3 (T ₃)		Anterior impact Test 4 (T ₄)		
	Mean	±SD	Mean	±SD	Mean	±SD	Mean	±SD	
Bone length (mm)	202.0	10.3	197.7	9.6	198.7	9.1	198.7	13.1	0.800 ^A
Cross-sectional moment of inertia (mm ⁴)	12,542.2	1765.9	12,788.5	2182.1	14,558.4	3781.3	11,467.8	2223	0.060 ^A
Number of fracture lines	4.5	1.3	2.9	1.7	3.8	1.3	3.6	0.9	T ₁ vs. T ₂ ^K
Fracture line/polygon length (mm)	265.7	127.7	165.7	92.8	177.8	123.6	176.6	71.5	0.140 ^A
Chip size (mm)	33.5	13.8	0	0	33.9	16.3	48.9	7.7	0.074 ^T

Ilustración 45. Estadísticas de características de una fractura

El número de líneas de fractura hace referencia a cualquier línea de fractura que tenga al menos **1 centímetro**. Pueden tener apariencias muy diversas (longitudinales, oblicuas, transversal o poligonal).

- Longitudinal: se trata de una línea recta que recorre cercano o paralelo el eje del hueso. Puede aparecer en todos los aspectos del hueso, pero normalmente se encuentran en el área de impacto o en el área contralateral, o en ambos. Se suele estudiar su presencia, número y localización.
- Oblicuas: son líneas de fractura que recorren con un ángulo a lo largo del eje del hueso. El área que queda entre dos líneas oblicuas se denomina polígono.
- Transversal: es una línea horizontal que rodea el hueso.

En cuanto al fragmento, que es la parte perdida en el punto de impacto, se estudia su presencia y el tamaño (longitud de circunferencial en mm).

Ejemplo de patrón de fractura creado en comparación a las estadísticas relativas a ese aspecto:

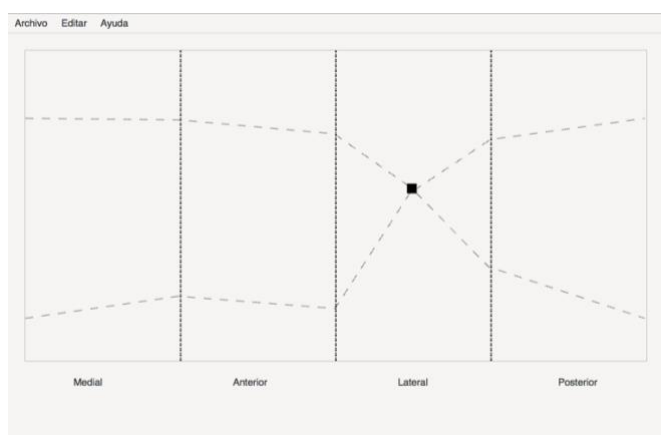


Ilustración 46. Patrón de fractura generados

Características según la dirección de impacto				
	Fractura actual	Estadísticas		
Tipo de fractura	Lateral	Valor medio	+/- SD	
Número de líneas de fractura	4	4,500000	1,300000	✓
Longitud de las líneas de fractura	705,084473	265,700000	127,700000	Hay demasiada variación, puede ser correcto
Tamaño del fragmento	25,736842	33,500000	13,800000	✓

Ilustración 47. Comparativa con respecto al primer test

En la ilustración anterior, se puede observar que, aunque el patrón cumpla algunas de las restricciones de que una fractura sea válida, existen otras que no, como el tamaño del fragmento para ese tipo de fractura (fractura lateral), así como longitud de las líneas de fractura. El hecho de que la longitud medida sea menor, se ve claramente que hay una gran variación entre longitudes en las distintas medidas del artículo, por ello nuestra medida también podría ser válida.

El hecho de que no haya pasado los test no quiere decir que el patrón de fractura esté necesariamente mal, ya puede ser que sea un caso aislado. Por ello, es necesario que el usuario tenga criterio para saber si los resultados obtenidos son o no coherentes, así como sí quiere modificar el patrón o no.

- **Distribución de la localización de las líneas longitudinales según la dirección de impacto.**

Impact direction	Location of longitudinal lines (aspect)			
	Impact	Contra lateral	Impact + contra	Anterior
Lateral impact (Test 1)	25%	50%	0	25%
Posterior impact (Test 2)	34%	0	66%	0
Medial impact (Test 3)	57.5%	14%	28.5%	0
Anterior impact (Test 4)	72%	14%	14%	0

Ilustración 48. Localización de líneas longitudinales según la dirección de impacto

El análisis de la ilustración anterior calcula los valores en función de un grupo de 10 impactos, que da lugar a un porcentaje de líneas longitudinales en cada uno de los aspectos de un hueso, obteniendo los porcentajes de líneas longitudinales en el lugar del impacto, en la cara contralateral, en la cada del impacto + contralateral y en el aspecto anterior, dependiendo del tipo de fractura que se está estudiando (impacto lateral, impacto posterior, impacto medial e impacto anterior).

Todos los valores dados en la ilustración son calculables en la aplicación, por lo que podemos tener un estudio intensivo de la fractura y obtener unos resultados completamente analizables con respecto a esta tabla de estudio, aunque al ser los porcentajes un estudio de varias ejecuciones, solo podemos asegurar en qué aspecto no pueden aparecer líneas longitudinales.

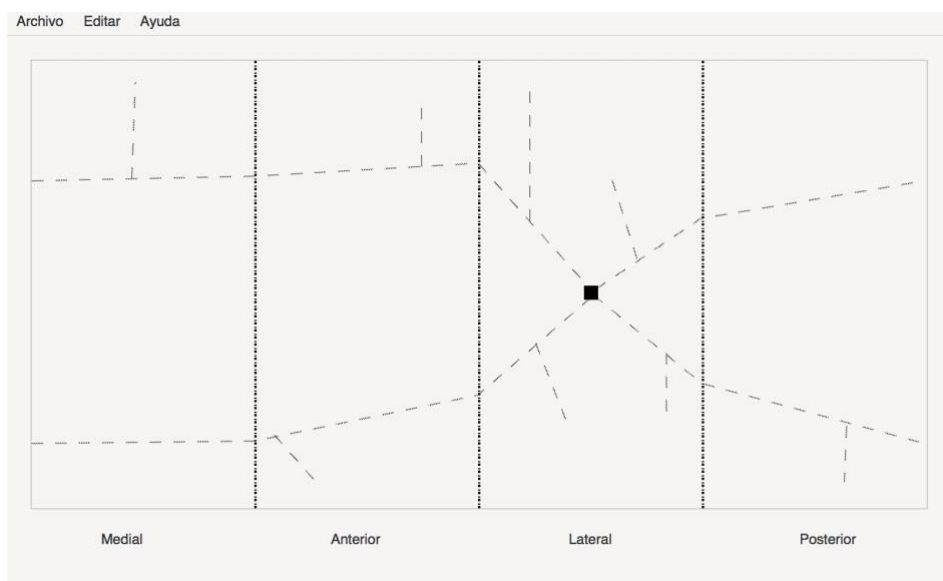


Ilustración 49. Fractura lateral con test 2

Líneas longitudinales según la dirección de impacto

En fracturas de impacto lateral no se ha obtenido en ninguno de los test realizados en el artículo de la revista Forensic Science ninguna caso en el cuál haya líneas longitudinales tanto en el aspecto de impacto como en el contralateral, por lo que su fractura podría estar errónea

Ilustración 50. Análisis de fractura lateral con test 2

Fijándonos en los resultados, podemos ver claramente que no es válida dicha fractura, por ello, podemos hacer también otro test, como el del apartado anterior, para ver si vuelve a fallar y así corroborar que realmente no es una fractura válida.

También podemos encontrar otro tipo de mensaje que hace referencia a que nuestra fractura no presenta nada raro en la posición de las líneas respecto a las líneas del estudio de referencia.

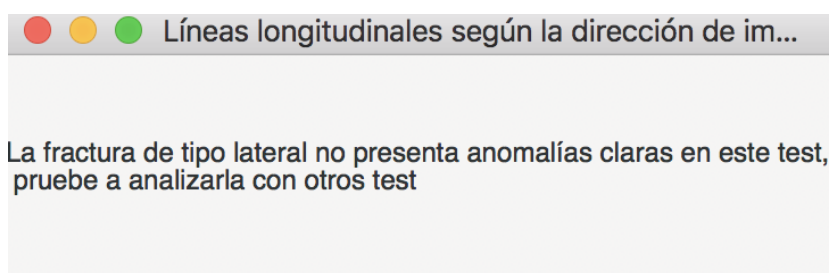


Ilustración 51. Test 2 válido

- Media de líneas de fractura según la dirección de impacto

Impact direction	Bone aspect								P* value
	Lateral		Anterior		Posterior		Medial		
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	
Lateral impact	4.3	1.2	2.5	0.7	2.2	0.6	2.7	0.9	0.001
Anterior impact	1.8	0.8	4.2	0.8	1.9	0.6	2	0.7	<0.001
Medial impact	2.3	0.7	2	0.7	1.3	0.5	2.9	1.5	0.004
Posterior impact	1.4	0.52	2.4	1.1	2	1	1.3	0.5	0 05

Ilustración 52. Media de líneas de fractura según dirección de impacto

En este caso, lo que se analiza es el número de líneas de fractura que hay en cada uno de los aspectos en los distintos tipos de fracturas y la desviación que pueden tener dichos valores, dentro de los cuales sería un patrón de fractura válido.

Cogiendo otro ejemplo de patrón de fractura creado y calculando sus test, obtenemos lo siguiente:

Nº de líneas de fractura según la dirección del impacto				
Fractura actual		Estadísticas		
Tipo de fractura	Lateral	Valor medio	+/- SD	
Medial	2	2,700000	0,900000	✓
Anterior	2	2,500000	0,700000	✓
Lateral	4	4,300000	1,200000	✓
Posterior	2	2,200000	0,600000	✓
La validación positiva indica que está dentro del 80% de los resultados dados por test				
La validación negativa indica que está fuera del 80% de los resultados dados por test				

Ilustración 53. Número de líneas medio por aspecto según dirección de impacto

En la imagen se ve claramente que pasaría este test, por lo que se podría decir que la fractura lateral realizada es correcta, ya que todos los valores están dentro del 80% de los resultados dados por el test.

De acuerdo a todos los datos que la aplicación es capaz de recoger y analizar los datos, podemos afirmar que un patrón de fractura realizado es o no es válido de acuerdo a los datos recogidos en los distintos artículos estudiados.

4.3. Validación por expertos

Para obtener una validación cualitativa de la calidad de la fractura, se necesita de un conjunto de valoraciones por parte de los expertos del patrón de fractura generado. En la búsqueda de la mayor fiabilidad posible de nuestro patrón de fractura, se ha creado un cuestionario que han realizado los distintos expertos para validar dicha fractura, teniendo en cuenta los datos de una fractura real, las estadísticas y su experiencia en el área. En dicho cuestionario también se ha incluido una valoración de la herramienta, incluyendo aspectos como fluidez, facilidad de uso, que sea intuitiva, el feedback obtenido, etc.

A partir del análisis de dichos resultados se ha obtenido la validez de la herramienta, revelando si realmente los patrones realizados por los usuarios con la ayuda de la herramienta son válidos o no y la calidad de la herramienta.

4.3.1. Validación de la herramienta

El test de validación de la herramienta tomará una muestra de dos expertos únicamente para realizar los test y analizar los resultados con el fin de determinar una validez aproximada del software, debido a que es difícil contar con especialistas que sepan de este tema o sean próximos a esta temática.

Los resultados recogidos de las encuestas son los siguientes:

Pregunta	1	2	Media
Edad	28	25	26.5
Experiencia	Alta	Media	Media/alta
1. Encuentro la herramienta fácil de usar	4	4	4
2. Requerí de ayuda para realizar las tareas solicitadas	4	4	4
3. Considero al producto fácil de usar para cualquier persona con conocimientos en el área y con capacidad de manejar una computadora o sistema	3	5	4
4. Mensajes e instrucciones ambiguos o incorrectos	5	5	5
5. Número de pasos excesivos para realizar una tarea	5	5	5
6. Considero seguro al producto software	5	4	4.5
7. Cometí uno/muchos error/errores durante la realización de las tareas, a causa de no leer claramente un mensaje o indicación del producto software	5	3	4
8. Cometí uno/muchos error/es durante la realización de las tareas a causa de que algún/os mensaje/s o indicación/es de la herramienta software no estaban claros	5	4	4.5
8. El producto software tiene mensaje o instrucciones que no son claros para poder llevar a cabo una tarea	5	4	4.5
9. Considero elegante o atractivo al producto software (combinación de colores adecuada, tamaño de letra legible, etc.)	5	4	4.5
10. Estoy cómodo/a con el software	4	4	4
11. El modo de representación del patrón de fractura una vez acabado se corresponde con el modo de representación habitual de patrones en esta área de investigación.	5	5	5
12. Los pasos a seguir para realizar el dibujado de un patrón son intuitivos	3	5	4
13. Los pasos necesarios para crear un fragmento son intuitivos	5	5	5
14. Los pasos necesarios para modificar un fragmento son sencillos	5	4	4.5
15. Los pasos necesarios para ajustar las características de una línea de fractura son sencillos	5	4	4.5
16. La secuencia de pasos para dibujar una línea oblicua son los necesarios	3	4	3.5

17. En general, el dibujado de las líneas de fractura es intuitivo y cómodo de realizar	5	5	5
18. He necesitado la pestaña de ayuda de la herramienta para saber comenzar a usar la herramienta	5	4	4.5
19. He necesitado la pestaña ayuda mientras usaba la herramienta	4	3	3.5
20. Grado de satisfacción con la herramienta	5	5	5
21. Volveré a usar la herramienta si tengo que hacer una tarea relacionada con lo ofrecido en el prototipo	4	5	4.5
22. En cuanto a la validación de fractura... a. El sistema usado ha valorado aspectos en el patrón como correcto y bajo mi experiencia la fractura es correcta. b. El sistema usado ha valorado los aspectos en el patrón como correcto y bajo mi experiencia en el área la fractura es incorrecta. c. El sistema usado ha valorado aspectos en el patrón como incorrectos y bajo mi experiencia la fractura es correcta. d. El sistema usado ha valorado aspectos en el patrón como incorrectos y bajo mi experiencia la fractura es incorrecta	A	A	A
23. ¿Cree que debería ser más guiada la generación del patrón? a. Sí b. No	A	B	50 % requiere ser mejor guía y el 50% no lo requiere

Tabla 7. Resultados test de expertos

En la encuesta se puede ver que los usuarios son jóvenes y están habituados a este tipo de patrones de fractura al tener una experiencia media/alta.

Por otro lado, se ve claramente que los usuarios encuentran la herramienta fácil de usar, por los resultados en la primera pregunta, pero que necesitan de ayuda la primera vez de uso para saber usarla, ya sea ayuda personal o con la pestaña de ayuda, como se ve en la segunda pregunta, 19 y 20. Además, creen que cualquier persona con conocimientos en el área y capacidad de manejar un ordenador puede

usar la herramienta de forma sencilla (pregunta 3), por lo que podemos ver aunque ellos mismo hayan necesitado de la pestaña de ayuda de la herramienta puede ser solamente por la primera entrada de contacto con la herramienta y en un segundo uso la consideren totalmente usable sin necesidad de la misma.

En cuanto a corrección de la herramienta, desde la pregunta 4 a la 9, podemos ver que la herramienta es fiable, ya que se mueve en un intervalo de puntuación 4.5 – 5.0, por lo que se podría decir que funciona correctamente.

Además. la interfaz les parece adecuada (pregunta 10, 11), así como el modo de representación del patrón en el canvas de dibujado (pregunta 12).

En lo referente al dibujado, de la pregunta 13 a la 18, se puede considerar que la forma de dibujado es intuitiva y correcta, ya que la media de resultados de estas preguntas es de una puntuación de 4.416.

Finalmente, se puede concluir que la herramienta es adecuada ya los expertos están satisfechos con la misma y volverían a usarla, ya que al hecho de que la herramienta funcione correctamente, se le añade que, analizando la corrección de la fractura generada por ellos mismos, les parece correcta y válida. Por lo que podemos concluir diciendo que el editor de fracturas cumple sus objetivos en los que al dibujado se refiere y que no es complicado de usar.

4.3.2. Validación de los patrones de fractura

Tras analizar los distintos tipos de test realizados por los expertos, queda la última prueba, que trata de someter a los expertos a una prueba de análisis de patrones creados con la herramienta software desarrollada en este proyecto. Dichos expertos tendrán que verificar si son correctas o no y exponer los motivos por los cuales es dicho resultado su valoración.

En primer lugar, cabe destacar que únicamente se van a poner dos de los análisis realizados a modo de ejemplo, para ver dos opiniones de expertos sobre los mismos patrones y que las distintas pruebas han sido realizadas a los expertos sin la presencia del resto de expertos para que no se vean contaminados por la opinión de los demás.

a) Patrón de fractura 1

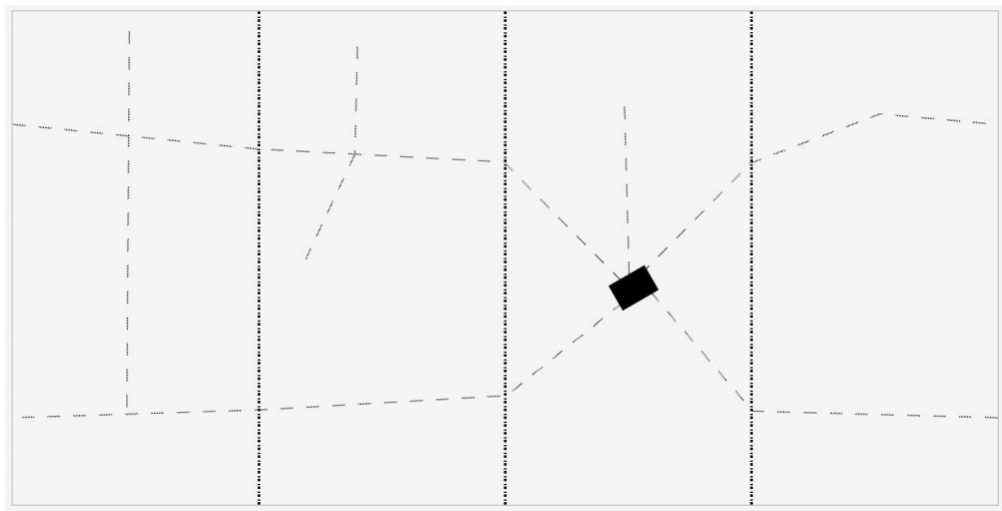


Ilustración 54. Imagen 1 a valoración de expertos

Experto 1

Encuentra que el patrón es correcto debido a que hay fragmento en el aspecto lateral y no es muy grande. Además, recalca que hay pocas líneas en este tipo de fracturas, por lo que lo considera correcto.

Experto 2

Encuentra correcta al mirar la fractura, no argumenta por qué, pero sabe que lo es.

b) Patrón de fractura 2

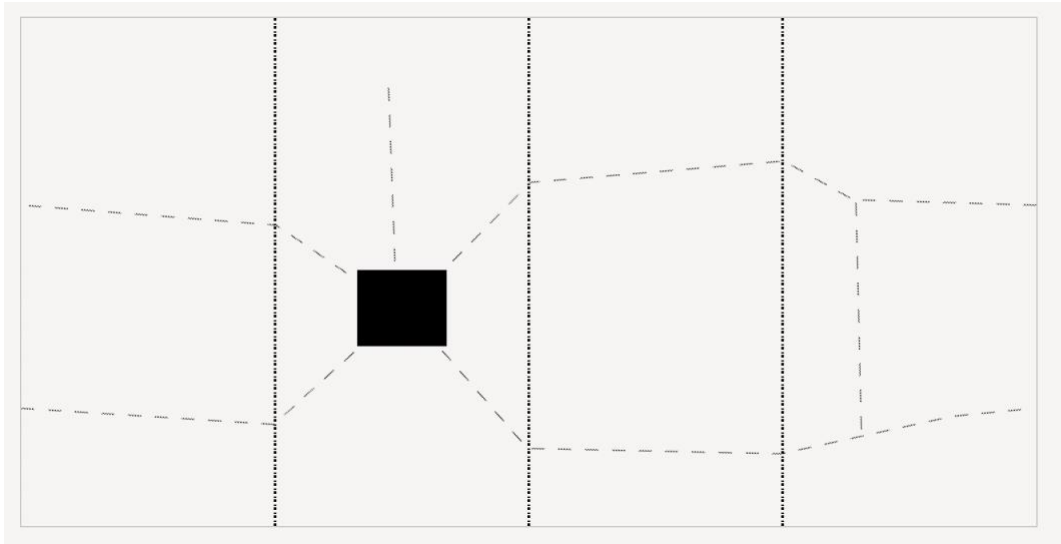


Ilustración 55. Imagen 2 sometida a valoración experta

Experto 1

Encuentra que el patrón es incorrecto debido a que el fragmento es muy grande, si no fuera así, encontraría la fractura correcta.

Experto 2

Encuentra incorrecto el patrón al ser el fragmento muy grande.

c) Patrón de fractura 3

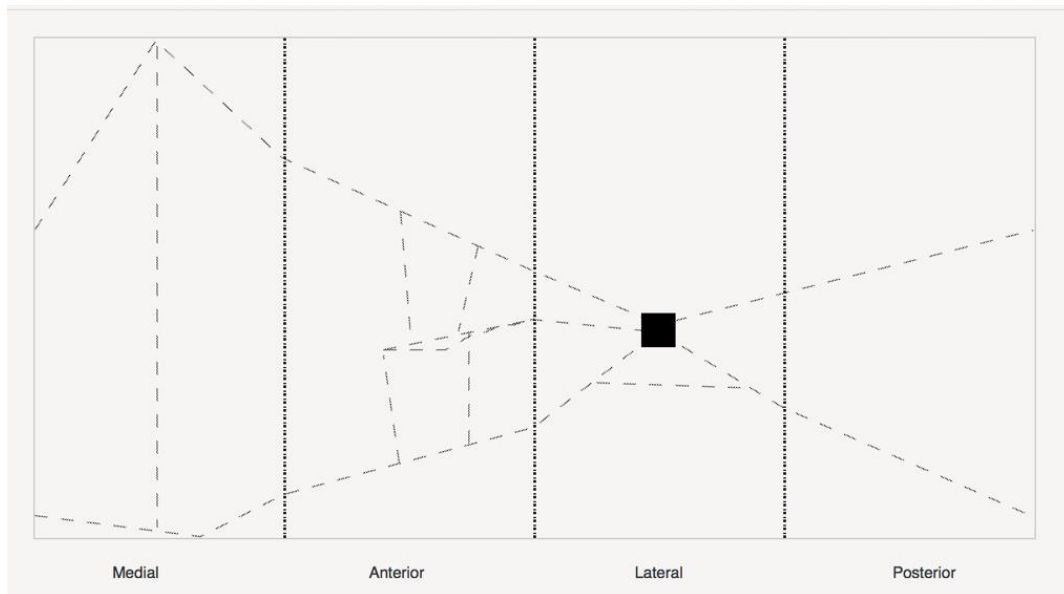


Ilustración 56. Imagen 3 sometida a valoración experta

Experto 1

Encuentra que el patrón es incorrecto debido a los ángulos de inclinación y las ramificaciones dentro de ramificaciones.

Experto 2

Encuentra correcto al patrón, aunque echa en falta más ramificaciones en el aspecto posterior.

d) Patrón de fractura 4

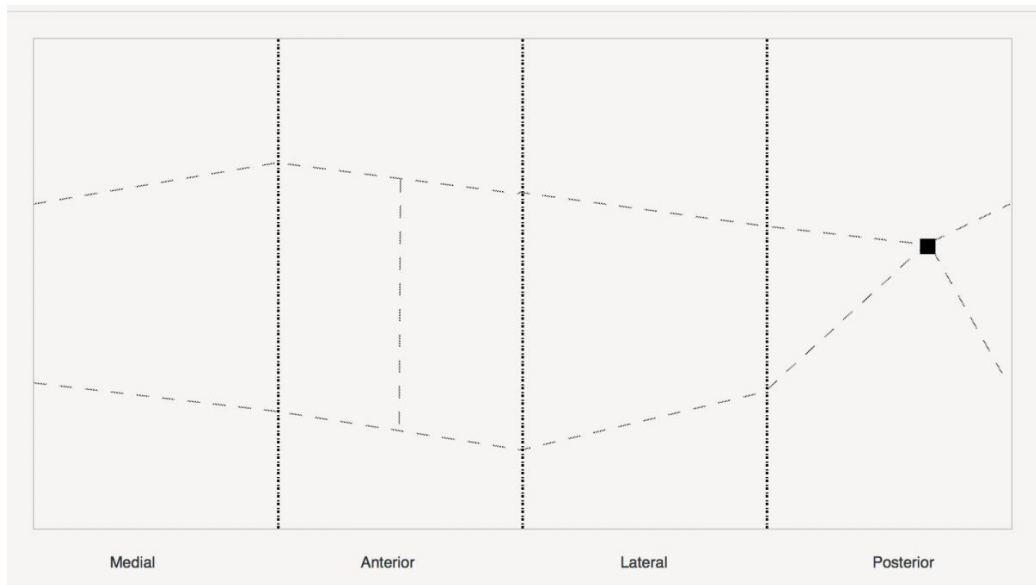


Ilustración 57. Imagen 4 sometida a valoración experta

Experto 1

Encuentra que el patrón es incorrecto debido que tiene fragmento y líneas de fractura longitudinales.

Experto 2

Encuentra incorrecto al patrón debido al tamaño del fragmento.

e) Patrón de fractura 5

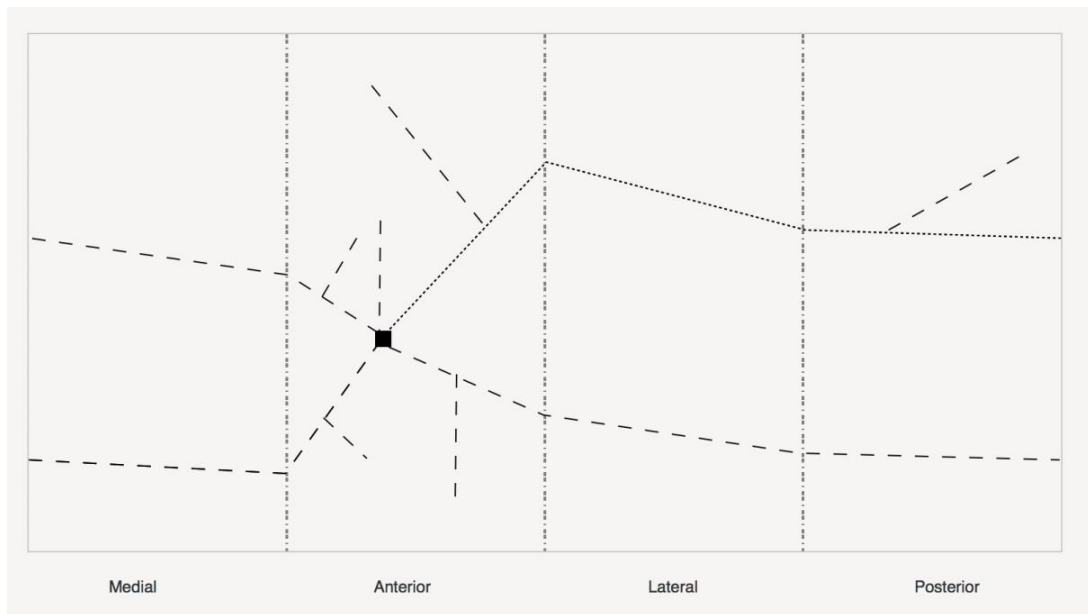


Ilustración 58. Imagen 5 sometida a valoración experta

Experto 1

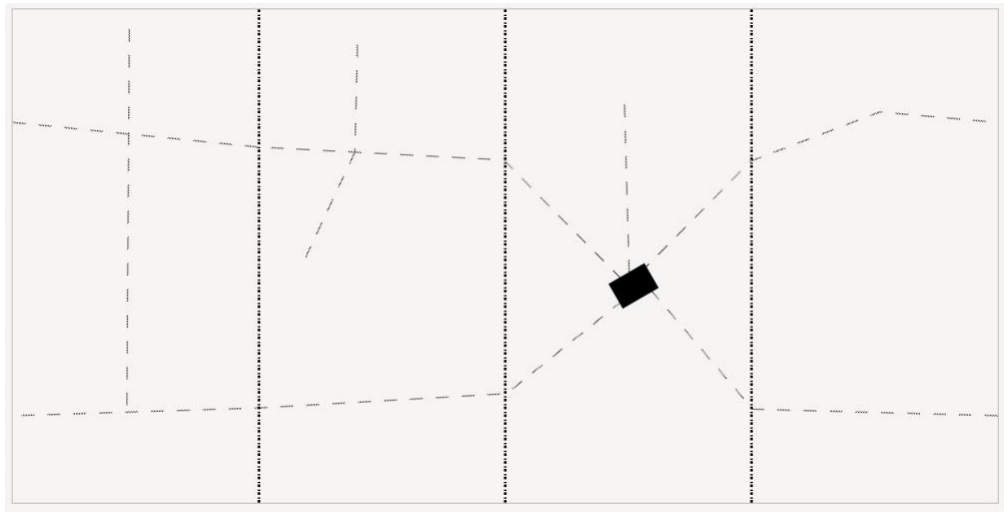
Encuentra que el patrón es correcto.

Experto 2

Encuentra el patrón de fractura correcto.

5. Resultados y discusión

En este apartado, basándonos en los resultados del apartado anterior, vamos a realizar una comparativa entre los resultados obtenidos por los expertos y los obtenidos por la propia herramienta. Para ello, se va a indicar la valoración automática en cada uno de los patrones y, a continuación de cada valoración automática, se compararán los resultados de los expertos y los resultados de la herramienta. Finalmente, una vez se hayan obtenido los resultados para todos los patrones se procederá a una conclusión general.

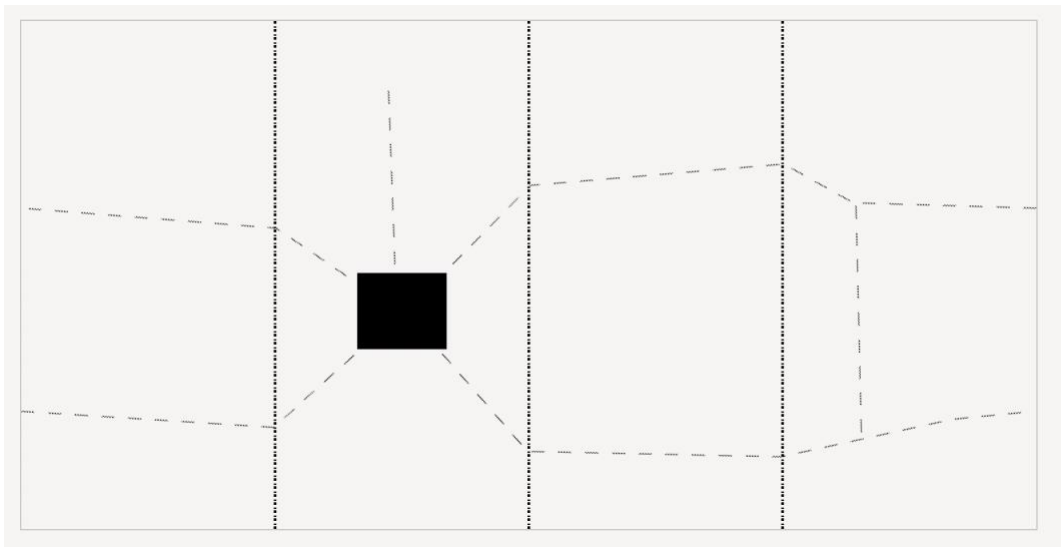
a) Patrón de fractura 1**Ilustración 59. Imagen 1 a valoración de expertos**

Según la herramienta de validación automática, para la primera validación, en la que se analizan las características de la fractura según la dirección de impacto, encontramos que son correctos los tres aspectos estudiados: número de líneas de fractura, longitud de las líneas de fractura y tamaño del fragmento.

En el segundo test, que analiza las líneas longitudinales, obtenemos que el patrón podría tener algún error al no corresponder completamente en las estadísticas, pero que para más seguridad se deberían seguir realizando más test.

En el tercer y último test de validación, se analizan las líneas de fractura existentes en cada aspecto impactado, encontrando que en todos los casos se encuentra dentro del 80% de los resultados dados por las estadísticas, encontrando válido el patrón.

Por tanto, realizando un análisis general de los resultados de la herramienta, se puede ver que tanto los expertos como validación automática concluyen con el mismo resultado en este primer patrón.

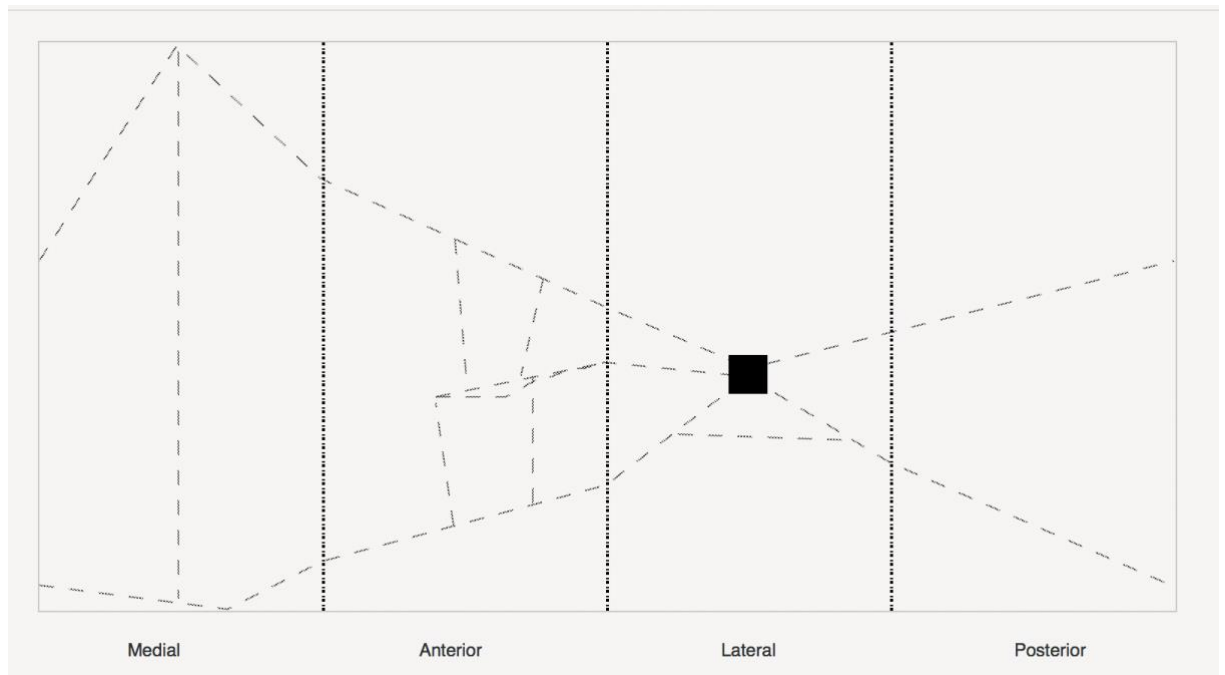
b) Patrón de fractura 2**Ilustración 60. Imagen 2 sometida a valoración experta**

Según la herramienta de validación automática, para la primera validación, en la que se analizan las características de la fractura según la dirección de impacto, encontramos que son correctos dos de los tres aspectos estudiados: número de líneas de fractura, longitud de las líneas de fractura, pero no el tamaño del fragmento.

En el segundo test, que analiza las líneas longitudinales, obtenemos que el patrón no presenta ningún tipo de anomalía en cuanto a las líneas longitudinales, por lo que deberíamos seguir realizando los demás test para una mejor valoración.

En el tercer y último test de validación, se analizan las líneas de fractura existentes en cada aspecto impactado, encontrando que en todos los casos se encuentra dentro del 80% de los resultados dados por las estadísticas, exceptuando el aspecto posterior, que se encuentra fuera de este rango.

Por tanto, realizando un análisis general de los resultados de la herramienta, se puede ver que tanto los expertos como la validación automática concluyen con el mismo resultado en este primer patrón, el patrón es incorrecto.

c) Patrón de fractura 3**Ilustración 61. Imagen 3 sometida a valoración experta**

Según la herramienta de validación automática, para la primera validación, en la que se analizan las características de la fractura según la dirección de impacto, encontramos que son correctos dos de los tres aspectos estudiados: número de líneas de fractura, longitud de las líneas de fractura, pero no el tamaño del fragmento.

En el segundo test, que analiza las líneas longitudinales, obtenemos que el patrón no presenta ningún tipo de anomalía en cuanto a las líneas longitudinales, por lo que deberíamos seguir realizando los demás test para una mejor valoración.

En el tercer y último test de validación, se analizan las líneas de fractura existentes en cada aspecto impactado, encontrando que en todos los casos se encuentra dentro del 80% de los resultados dados por las estadísticas.

Por tanto, realizando un análisis general de los resultados de la herramienta, se puede ver que en un tipo de patrón de fractura más complicado los mismos expertos difieren en sus resultados y finalmente es la herramienta de

validación automática quien inclina la balanza dejando ver que el patrón de fractura sería correcto si el tamaño del fragmento fuera un poco más pequeño, ya que la combinación de las líneas en los distintos aspectos es correcta.

d) Patrón de fractura 4

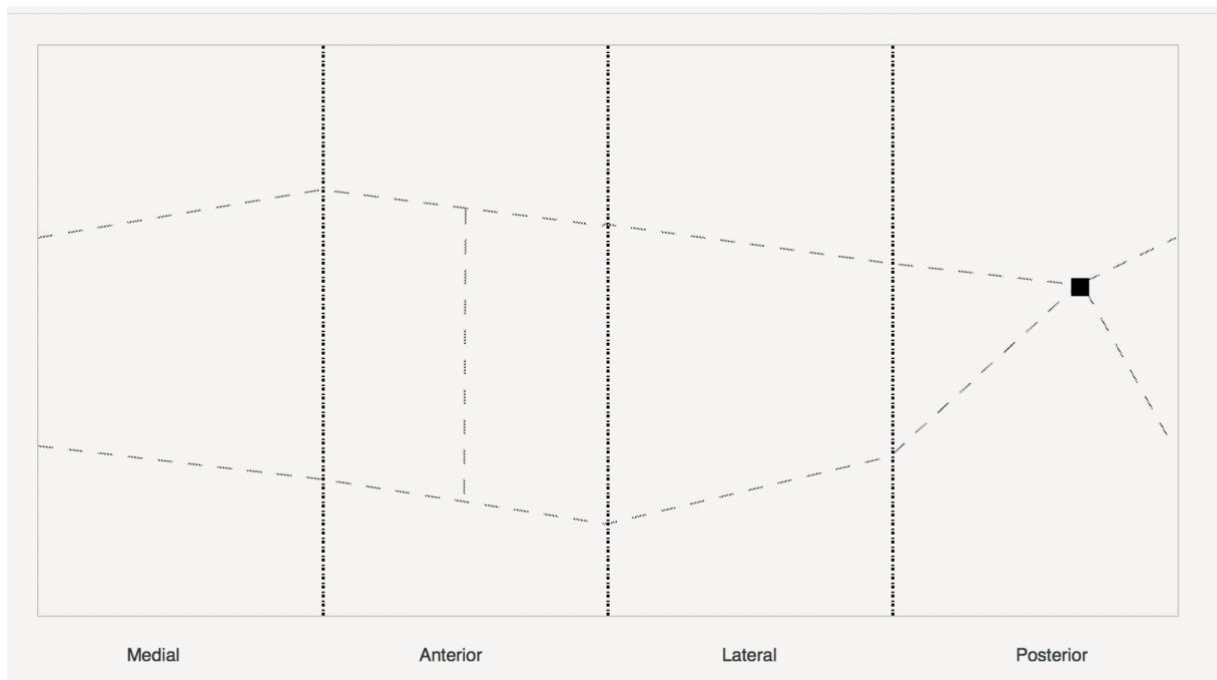


Ilustración 62. Imagen 4 sometida a valoración experta

Según la herramienta de validación automática, para la primera validación, en la que se analizan las características de la fractura según la dirección de impacto, encontramos que son correctos dos de los tres aspectos estudiados: número de líneas de fractura, longitud de las líneas de fractura, pero no el tamaño del fragmento.

En el segundo test, que analiza las líneas longitudinales, obtenemos que el patrón no presenta ningún tipo de anomalía en cuanto a las líneas longitudinales, por lo que deberíamos seguir realizando los demás test para una mejor valoración.

En el tercer y último test de validación, se analizan las líneas de fractura existentes en cada aspecto impactado, encontrando que en todos los casos se encuentra dentro del 80% de los resultados dados por las estadísticas.

Por tanto, realizando un análisis general de los resultados de la herramienta, se puede ver que tanto los expertos como la herramienta de validación automática coinciden en que el tamaño del fragmento es incorrecto o no debería de estar ahí. Por tanto, el patrón es incorrecto.

e) Patrón de fractura 5

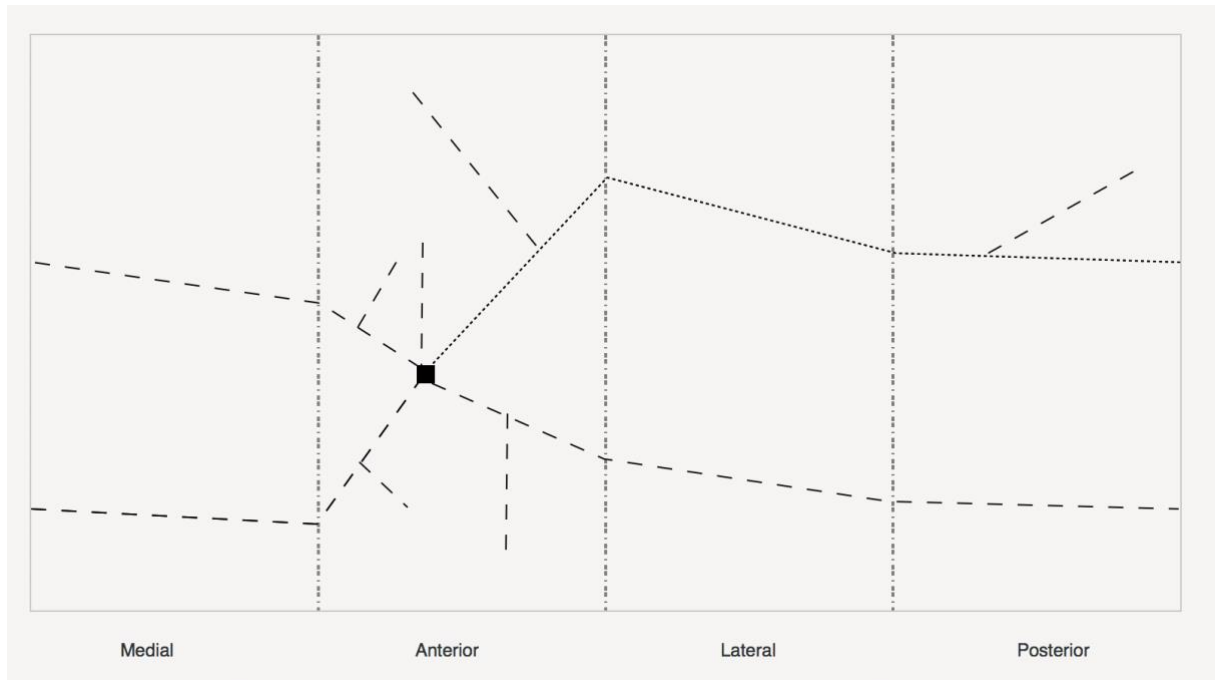


Ilustración 63. Imagen 5 sometida a valoración experta

Según la herramienta de validación automática, para la primera validación, en la que se analizan las características de la fractura según la dirección de impacto, encontramos que son incorrectos dos de los tres aspectos estudiados: número de líneas de fractura y tamaño del fragmento, aunque la longitud de las líneas de fractura es correcta.

En el segundo test, que analiza las líneas longitudinales, obtenemos que el patrón no presenta ningún tipo de anomalía en cuanto a las líneas longitudinales, por lo que deberíamos seguir realizando los demás test para una mejor valoración.

En el tercer y último test de validación, se analizan las líneas de fractura existentes en cada aspecto impactado, encontrando solo dos de los aspectos se

encuentran con un número de líneas de fractura dentro del 80% de los resultados dados por las estadísticas.

Por tanto, realizando un análisis general de los resultados de la herramienta, se puede ver que los expertos difieren en valoración respecto a la herramienta de validación automática. Por tanto, según la valoración automática, el patrón es incorrecto, ya que, según los estudios, en una fractura anterior no puede haber tantas líneas de fractura en la parte anterior en comparación con el resto de aspectos, además el tamaño del fragmento es incorrecto, por lo que la herramienta nos estaría diciendo un caso en el que la fractura es incorrecta estadísticamente y los expertos creen que es correcta. En este tipo de casos, habría que guiarse por la intuición de los expertos y concluir que podría ser correcta.

- Conclusión final

Fractura	Validación de experto 1	Validación de experto 2	Validación automática
Patrón de fractura 1	Correcto	Correcto	Correcto
Patrón de fractura 2	Incorrecto	Incorrecto	Incorrecto
Patrón de fractura 3	Incorrecto	Correcto	Correcto
Patrón de fractura 4	Incorrecto	Incorrecto	Incorrecto
Patrón de fractura 5	Correcto	Correcto	Incorrecto

Tabla 8. Valoración experta VS automática

Como conclusión final, comentar que realmente el validador automático da con la valoración correcta en 4 de 5 casos, siendo el quinto caso un poco ambiguo hasta para los propios expertos, por lo que realmente no se podría decir que no esté en lo correcto, por lo que se puede decir que la herramienta implementada para determinar la corrección del patrón de fractura es válida y que se puede confiar en sus resultados.

6. Conclusiones y aspectos de mejora

La realización de este trabajo ha sido motivadora, ya que el hecho de aplicar la informática en temas relacionados con el ámbito de la salud, como es el caso de las fracturas, me parece muy alentador al poder llevar esta ciencia a aspectos de la vida más humanitarios ayudando a las personas y facilitando las arduas tareas a las que se someten diariamente los trabajadores, como en el caso de los médicos con las reconstrucciones de fracturas.

Por ello, creo que este trabajo no debería quedar aquí, ya que puede seguir explotándose la herramienta en otros aspectos para que su función pueda seguir siendo más aún de ayuda. Por ejemplo, se podría realizar una exportación del patrón de fractura en forma de imagen y que se pudiera aplicar directamente un hueso 3D para que el creador del patrón pudiera ver cómo quedaría ese tipo de fractura y si era lo que realmente se buscaba y poder seguir analizando a partir de ahí la corrección de dicho patrón o simplemente realizar otra serie de tareas sobre el mismo que se crean convenientes.

Por otro lado, realmente no existen una herramienta de edición de patrones de fractura propiamente dicha a parte de la que se ha implementado en este proyecto. Si es cierto, que se está trabajando en el grupo de investigación de la Universidad de Jaén especializada en este campo en un generador automático de patrones de fractura, que es con el editor con el que se ha conseguido compatibilidad. Pero claro, en este tipo de generador de patrones no se puede configurar un patrón al gusto ni validarlo. Por tanto, los patrones creados en este generador automático podrán validarse en el editor de este proyecto. Uno de los pasos para dar en la mejora de esta herramienta, es ir añadiendo toda la información necesaria para realizar cualquier tipo de validación del patrón en combinación con las restricciones que se vayan añadiendo al generador automático, para poder analizar completamente todos los aspectos de los patrones a validar.

Un punto donde empezar sería que el editor permitiera realizar cualquier composición en un patrón de fractura, incluyendo nuevos tipos de fractura y

restricciones que puedan aparecer en nuevas publicaciones o investigaciones sobre los patrones de fractura.

Del mismo modo, se podría estudiar el patrón de fractura desde otro punto de vista para analizar la velocidad, aceleración y posición del elemento que realizó el impacto y así poder analizar otros aspectos que también pudieran tener interés.

También sería interesante incluir en la herramienta sugerencias automáticas de patrones de fractura válidos a medida que se va dibujando, según las características del patrón de fractura que se está generando o simplemente que permita una edición más libre de la fractura importada, ya que por el momento solo permite dibujar nuevas líneas o ir eliminando las líneas dibujadas en el orden en el que se encontraban en el fichero, al no tener información de conectividad entre unas líneas y otras en dicho fichero.

Finalmente, destacar que en un futuro se realizarán pruebas con más expertos de cara a la publicación de los resultados obtenidos en el desarrollo de este proyecto.

Bibliografía

- [1] B.-A. Guerin, Gestión de proyectos informáticos. Desarrollo, análisis y control, ENI, 2018.
- [2] «PCL,» [En línea]. Available: http://pointclouds.org/documentation/tutorials/pcl_visualizer.php. [Último acceso: Marzo 2019].
- [3] «Assimp,» Julio 2012. [En línea]. Available: http://assimp.sourceforge.net/lib_html/cmake_build.html. [Último acceso: Marzo 2019].
- [4] T. G. PROJECT, «GNOME,» [En línea]. Available: <https://wiki.gnome.org/Projects/GooCanvas>. [Último acceso: Julio 2019].
- [5] T. G. Project, «GNOME Developer,» 2005 - 2014. [En línea]. Available: <https://developer.gnome.org/gtk3/stable/>. [Último acceso: Julio 2019].
- [6] T. G. Project, «Glade Docs,» [En línea]. Available: <https://glade.gnome.org>. [Último acceso: Julio 2019].
- [7] «Eigen,» Doxygen, [En línea]. Available: <https://glade.gnome.org>. [Último acceso: Julio 2019].
- [8] R. S. Pressman, Ingeniería del Software. Un enfoque práctico, Mc Graw Hill.
- [9] «ECURED. Diagrama de Gantt,» [En línea]. Available: https://www.ecured.cu/Diagrama_de_Gantt. [Último acceso: Julio 2019].
- [10] «TecnoEmpleo,» [En línea]. Available: <https://www.tecnoempleo.com/informe-empleo-informatica.php>. [Último acceso: Julio 2019].
- [11] C. e. Al, «The effect of impact tool geometry and soft material covering on longitudinal fracture patterns in children,» *Int J Legal Med*, nº 131, pp. 1011 - 1021, 2017.
- [12] C. e. Al, «The influence of impact direction and axial loading on the bone fracture Pattern,» *Forensic Science International*, nº 277, pp. 197 - 206, 2017.
- [13] C. e. al, «The impact velocity and bone fracture pattern: Forensic perspective,» *Forensic Science International*, nº 266, pp. 54 - 62, 2016.
- [14] M. Cillero, «Diagrama de clases,» [En línea]. Available: <https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-clases/>. [Último acceso: Julio 2019].
- [15] «Mozilla, <<MDN web docs>>,» [En línea]. Available: https://developer.mozilla.org/es/docs/Glossary/Callback_function. [Último acceso: Julio 2019].
- [16] «Metodología Scrum,» [En línea]. Available: <https://blog.ida.cl/estrategia-digital/metodologia-scrum-en-proyectos-digitales/>. [Último acceso: Julio 2019].
- [17] «Generalidades de las fracturas,» [En línea]. Available: <https://www.efisioterapia.net/articulos/generalidades-las-fracturas>. [Último acceso: Julio 2019].
- [18] B. BONES. [En línea]. Available: http://kefren.ujaen.es:6905/?page_id=366. [Último acceso: Julio 2019].

Anexos

En este apartado se recogen una serie de documentos e información complementaria a la documentación.

- Anexo I: Manual de usuario
- Anexo II: Test de usabilidad
- Anexo III: Test de validación de expertos
- Anexo IV: Creación de un ejecutable con xCode e instalación en otro equipo

Se adjuntará a la documentación también un vídeo demostrativo de la aplicación, una imagen del diagrama de clases para una visión de la misma más detallada y una carpeta con el ejecutable del programa.

ANEXO I. Manual de usuario

Este manual de usuario se estructura en:

1. Introducción

2. Pasos para la creación de un patrón de fractura

- Paso 1. Dibujado del fragmento o punto de fractura
- Paso 2. Dibujado de las líneas de fractura
- Paso 3. Líneas longitudinales
- Paso 4. Líneas ramificadas de las líneas principales

3. Opciones de la herramienta gráfica

- Pestaña “Editar”. Deshacer operaciones
- Pestaña “Editar”. Rehacer operaciones
- Pestaña “Archivo”. Nuevo
- Pestaña “Archivo”. Abrir
- Pestaña “Archivo”. Guardar
- Pestaña “Archivo”. Guardar como
- Pestaña “Archivo”. Salir
- Pestaña “Ayuda”
- Pestaña “Ver”
- Pestaña Validar

1. Introducción

El editor de fracturas permite representar de forma fidedigna una fractura ayudándonos de un ratón.

En el dibujado de una fractura con el editor, hay que tener en cuenta que se realiza siguiendo un orden de dibujado, para que el usuario no se sienta abrumado por no saber seguir.

En primer lugar, deberá saber si quiere una fractura con desprendimiento de viruta o no. En caso de quererlo deberá elegir el tipo de fragmento, posicionarlo y a partir de ahí comenzar a dibujar las líneas de fractura. En caso de no querer que tenga fragmento de viruta, el usuario pasará al dibujado de las líneas, siendo el punto de impacto el lugar donde se clica por primera vez para dibujar la primera línea.

Por otro lado, el usuario deberá dibujar todas las líneas de fractura (suelen ser cuatro) y, una vez hecho esto, procederá al dibujado de las ramificaciones de las líneas de fractura.

Cuando el usuario considere que ha acabado de dibujar su fractura, podrá validarla, gracias al sistema de validación y que este sistema le diga si lo que ha dibujado se aproxima o no a una fractura real.

2. Pasos para la creación de un patrón de fractura

2.1. Paso 1: Dibujado del fragmento o punto de fractura

Para comenzar a dibujar una fractura, tenemos que saber antes los tipos de fractura que podemos encontrar y desde el punto de vista más general tenemos dos tipos de fracturas, fracturas con desprendimiento de un fragmento óseo o sin él y esto se refleja también en dicho editor.

Para que se pueda representar una fractura con fragmento, simplemente habrá que llevar el ratón hacia el apartado de fragmento y elegir el tipo de fragmento que se requiere para la fractura que se va a dibujar, pudiendo elegir

fragmento cuadrado, rectangular o circular. Una vez se elige el fragmento, se debe pulsar en la posición del canvas de dibujado en el que se quiere poner el fragmento, y este aparecerá ahí automáticamente.

Dicho fragmento se puede rotar y escalar, para que se tenga un control sobre el mismo y poder representar la fractura con la mayor flexibilidad posible.

En el caso de querer una fractura sin desprendimiento de un fragmento, simplemente, se pulsará sobre el canvas para comenzar a dibujar las líneas de fractura y se tomará como centro de impacto el punto donde se cliquee con el ratón en el canvas de dibujado

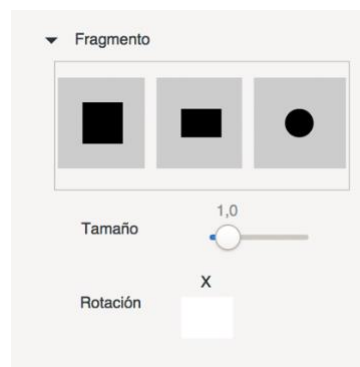


Ilustración 64. Configuración fragmento. Opciones

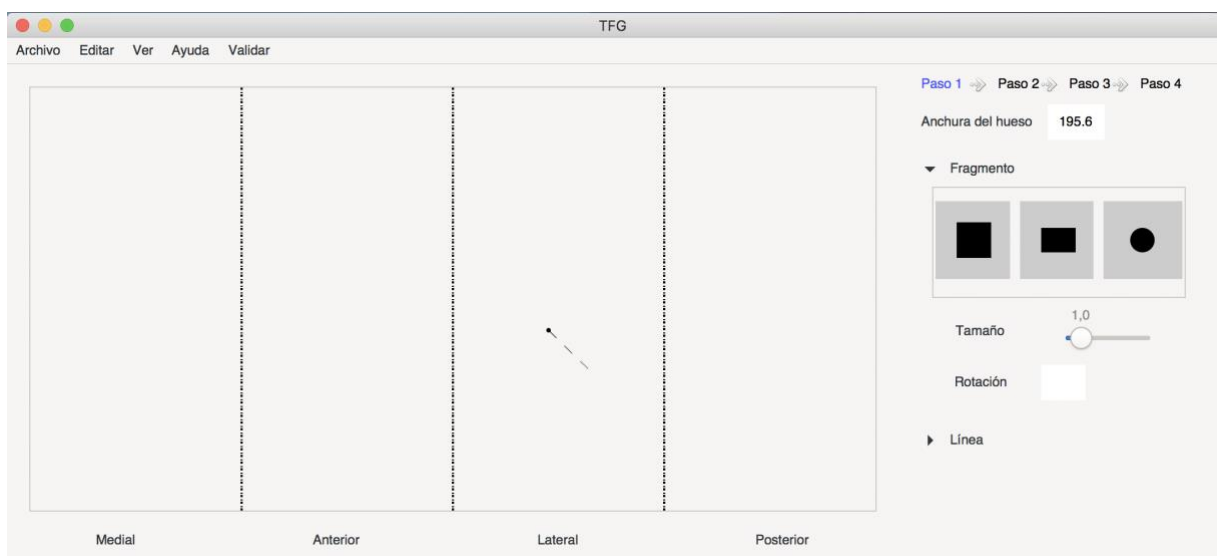


Ilustración 65. Imagen con punto de fractura.

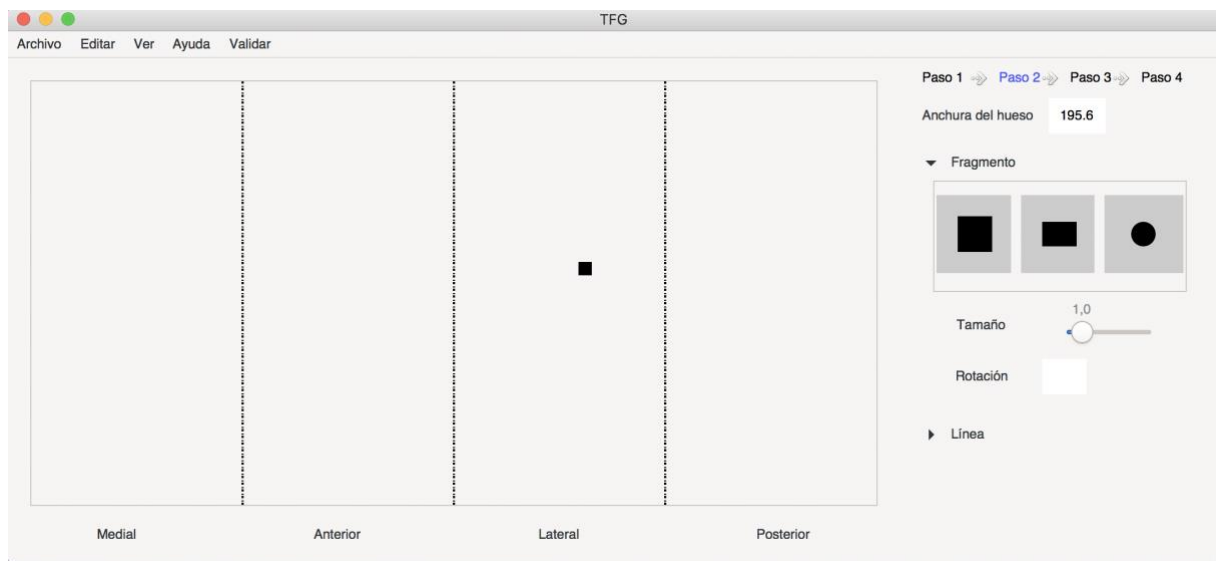


Ilustración 66. Patrón con fragmento de viruta

Una vez posicionado el fragmento o punto de fractura, se pasa al Paso 2 de forma automática.

2.2. Paso 2: Dibujado de las líneas de fractura

En el caso del dibujado de líneas de fractura, hay varias opciones de dibujado posibles, con el objetivo de darle la apariencia deseada a cada una de las líneas de fractura del canvas de dibujado.

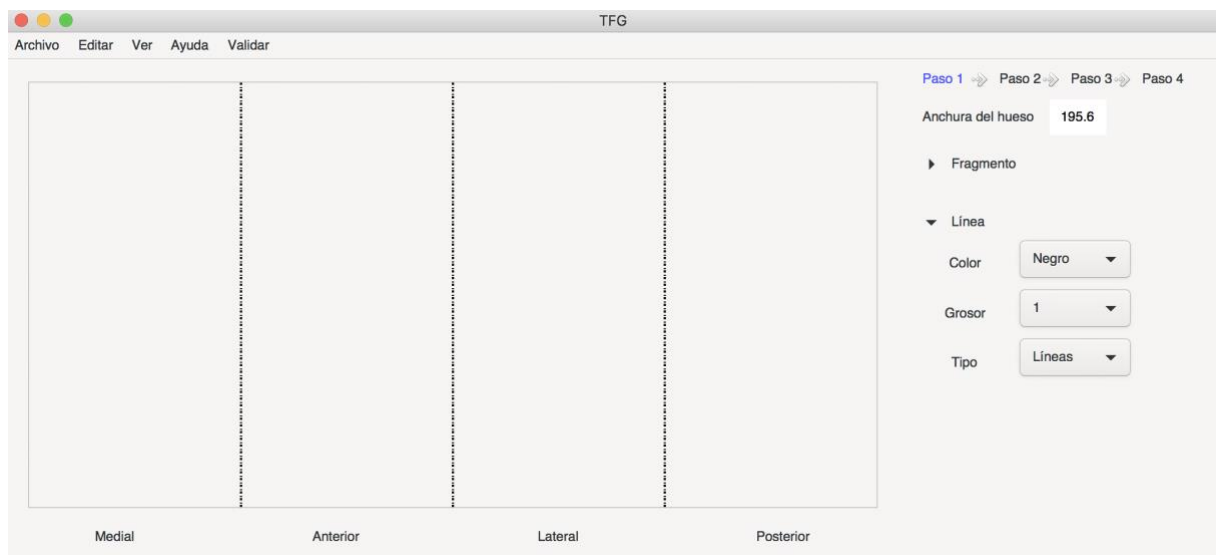


Ilustración 67. Configuración de una línea

Estas opciones de dibujado son las siguientes:

Tipo de línea:

- a) Línea sólida
- b) Línea punteada

Por defecto su valor es punteado.



Ilustración 68. Configuración de línea. Tipo

Grosor de la línea:

El grosor de la línea se puede escoger mediante un selector que permite coger desde el grosor 1, que sería la línea más fina posible, hasta un grosor de 3, siendo el máximo grosor disponible. Por defecto su valor es 1.

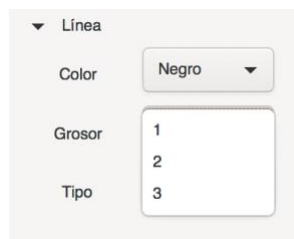


Ilustración 69. Configuración de línea. Grosor

Color de la línea

En cuanto al color de la línea es también elegible mediante un selector que nos permite poner las líneas en gris, rojo o verde. Por defecto su valor es negro.

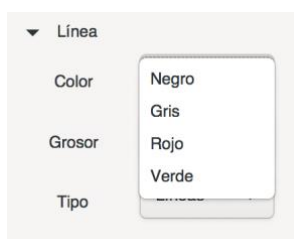


Ilustración 70. Configuración de línea. Color

Cabe destacar que todas estas opciones de dibujado de la línea pueden ser diferentes para cada una de las líneas o, si se quiere, que sea idéntica en todas ellas, para lo cual lo único que habría que hacer es configurar las características de las líneas nada más empezar o simplemente no configurarlo y escoger el tipo por defecto en caso de que nos guste.

Por otro lado, en el dibujado de las líneas de fractura hay varias formas de hacerlo, ya que se pueden encontrar líneas que salgan del fragmento o punto de fractura y se vayan extendiendo de forma horizontal con distinto ángulo de inclinación, llegando incluso a dar la vuelta completa al fragmento, o pueden ser ramificaciones de tamaño variable que aparecerán a partir de una línea de fractura principal.

En el dibujado de las ramificaciones tan solo habrá que hacer clic sobre la línea a partir de la cual queramos hacer una ramificación y mover el ratón hasta el lugar donde queramos que se quede dicha ramificación. Una vez hecho esto, tan solo hay que hacer clic con el botón izquierdo del ratón, logrando que la ramificación quede dibujada y fijada.

Para el dibujado de las líneas principales, se tendrán en cuenta algunas condiciones más.

En primer lugar, si se está dibujando una línea de fractura y dicha línea choca con una de las líneas de cuadrante del canvas, la línea acabará en dicho punto, aunque hagamos clic en el otro cuadrante, y comenzará de nuevo en ese punto y acabará en el siguiente cuadrante donde hicimos clic tras volver a realizarlo.

En segundo lugar, si el cuadrante con el que se choca se trata de los extremos del canvas, estaremos ante un caso de una línea que recorre todo el canvas de forma horizontal, por lo que la continuación de dicha línea está en el cuadrante correspondiente en cada caso. Si el choque se produce en el extremo derecho cuarto cuadrante, la continuación estaría en el primer cuadrante, y de producirse en el primer cuadrante a la izquierda la continuación sería en el cuarto

cuadrante. Para facilitar al usuario la continuación de dicha línea, aparece una línea guía en el cuadrante correspondiente que, al hacer clic sobre ella, desaparece y se continuaría dibujando la línea de fractura en el punto del cuadrante correspondiente donde se dejó, consiguiendo así una línea que pueda dar la vuelta completa al hueso.

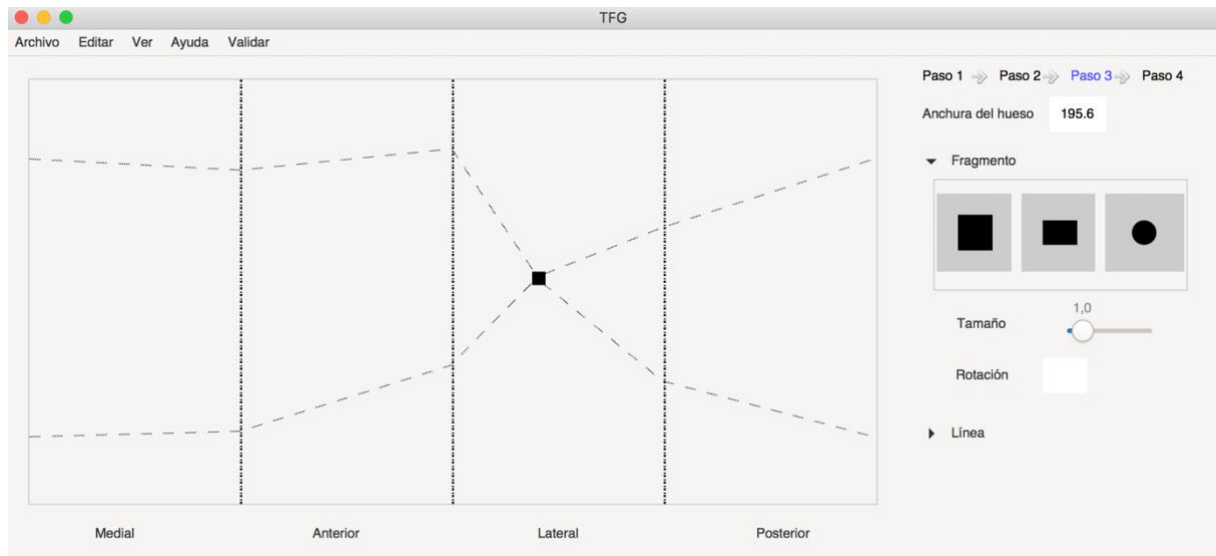


Ilustración 71. Patrón de fractura

Por último, habrá que tener en cuenta, que en las líneas de fractura existe una restricción de dirección entre el segmento de la polilínea que está dibujada hasta el momento y el siguiente segmento que se está dibujando, para evitar así que se pueda dibujar en sentido contrario al que va la línea, por lo que habrá casos donde no se pueda dibujar la línea de fractura elegida debido al incumplimiento de esta restricción, que haría no realista la fractura.

También se ha añadido otra restricción que trata de que las líneas de fractura no se entrecrucen, siendo solamente posible dibujar líneas de fractura que no sean secantes entre sí, a excepción de las ramificaciones de líneas de fractura, que si pueden entrecruzarse.

Además, las líneas de fractura principales se dibujan de forma continua, es decir, cuando se está dibujando una línea que atraviesa el canvas horizontalmente, tan solo habrá que ir haciendo clic en los puntos donde se

quiera poner un nuevo punto para la polilínea de la línea de fractura, pero tras hacer ese clic con el ratón se podrá seguir dibujando dicha línea sin tener que volver a pulsar sobre la misma.

Si se quisiera dejar de dibujar dicha línea, únicamente habría que presionar la tecla ESC del teclado. Si por el contrario lo que se quiere es seguir dibujando la línea de fractura que se acaba de terminar de dibujar mediante esta tecla o simplemente si se trata de una ramificación que se quiere continuar, simplemente habrá que hacer clic en el extremo de dicha línea.

2.3. Paso 3: Líneas longitudinales

Las líneas longitudinales se podrán dibujar cuando se esté en la fase 3 en todos los cuadrantes teniendo en cuenta que, en el aspecto impactado solo se puede dibujar sobre el fragmento y en el resto de aspectos si intersecta con alguna línea.

2.4. Paso 4: Líneas ramificadas de las líneas principales

Las ramificaciones son líneas que parten de otras líneas, para su dibujado es necesario pinchar con el ratón sobre una de las líneas y desplazar el ratón hasta la posición en la que queramos situar las líneas y volver a pinchar en dicha posición. Se pueden dibujar tantas como se desee.

3. Opciones de la herramienta

3.1. Pestaña “Editar”. Deshacer operaciones

Mediante la tecla CTRL+Z se pueden deshacer las operaciones realizadas sobre el canvas.

También se puede hacer mediante el botón deshacer de la pestaña “Editar” del editor de fracturas.

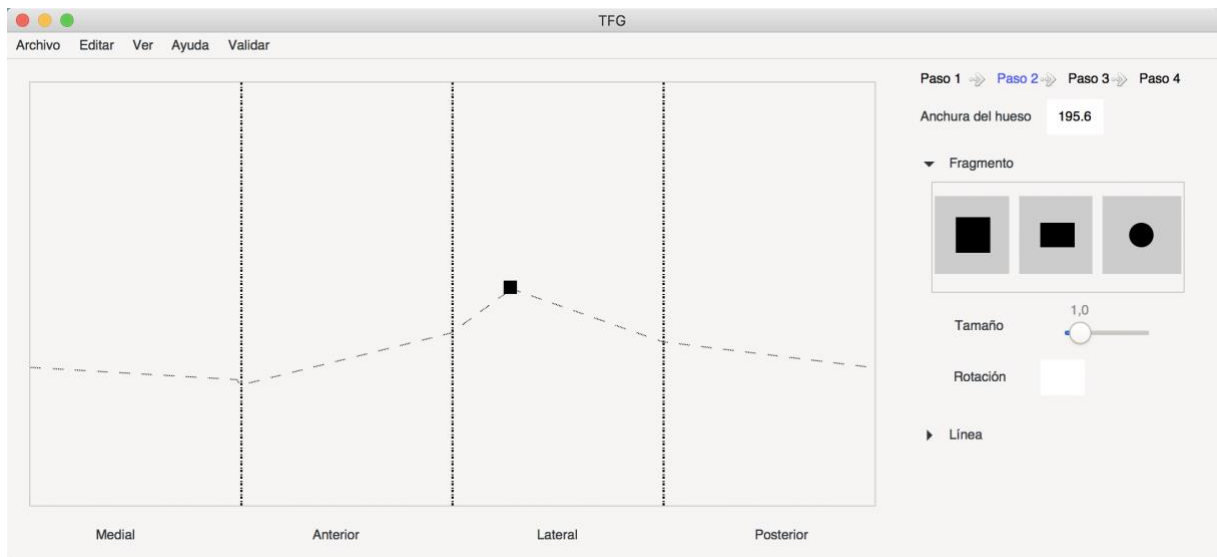


Ilustración 72. Antes de deshacer

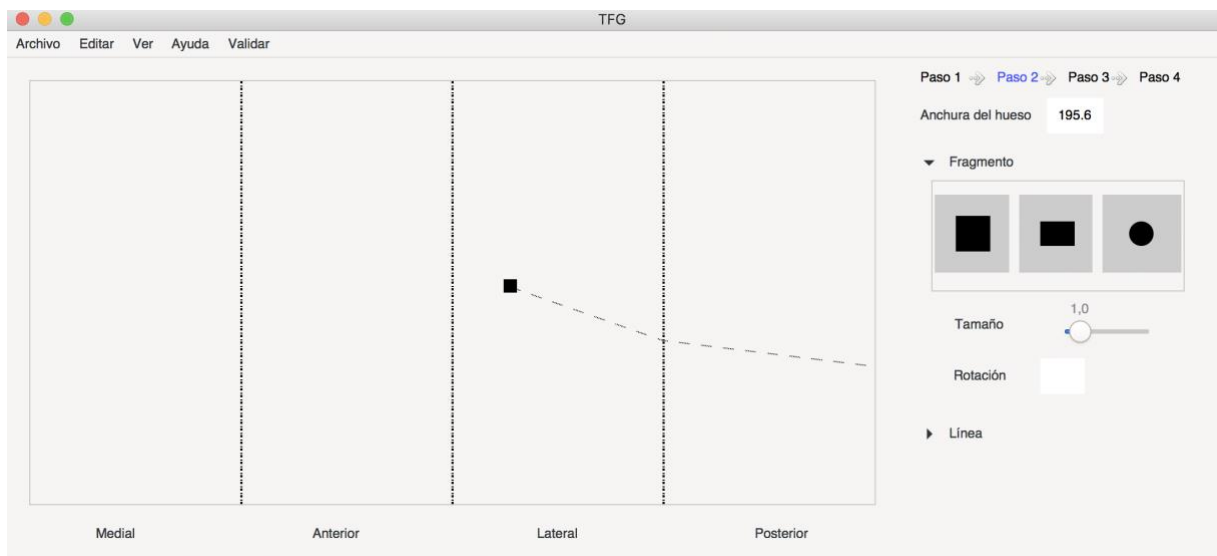


Ilustración 73. Operación deshecha

3.2. Pestaña “Editar”. Rehacer operaciones

Mediante la tecla CTRL+Y se pueden rehacer las operaciones realizadas sobre el canvas.

También se puede hacer mediante el botón rehacer de la pestaña “Editar” del editor de fracturas.

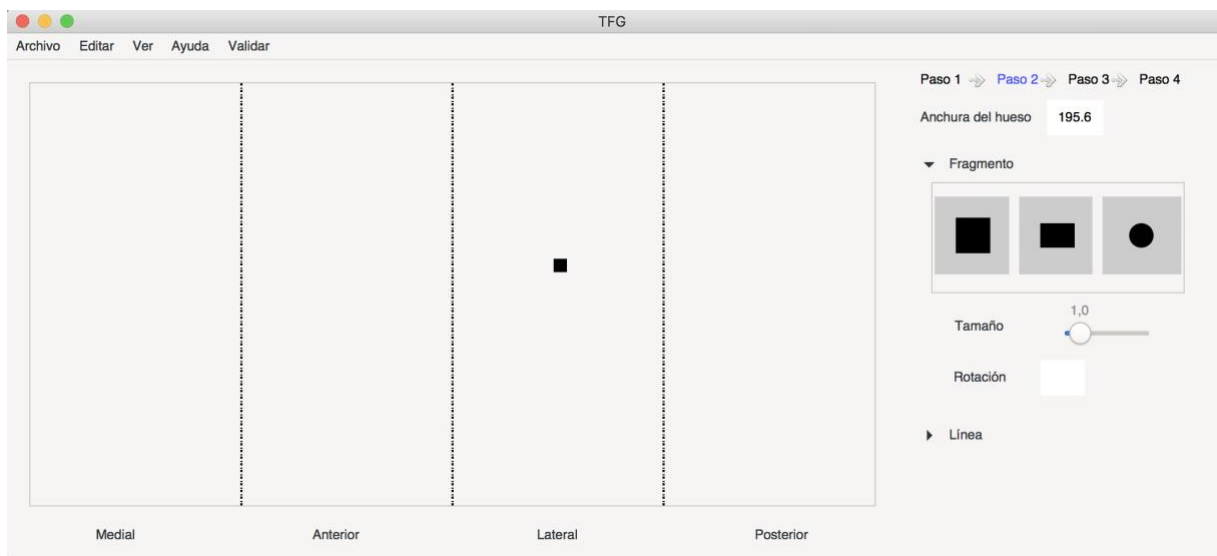


Ilustración 74. Operación deshecha

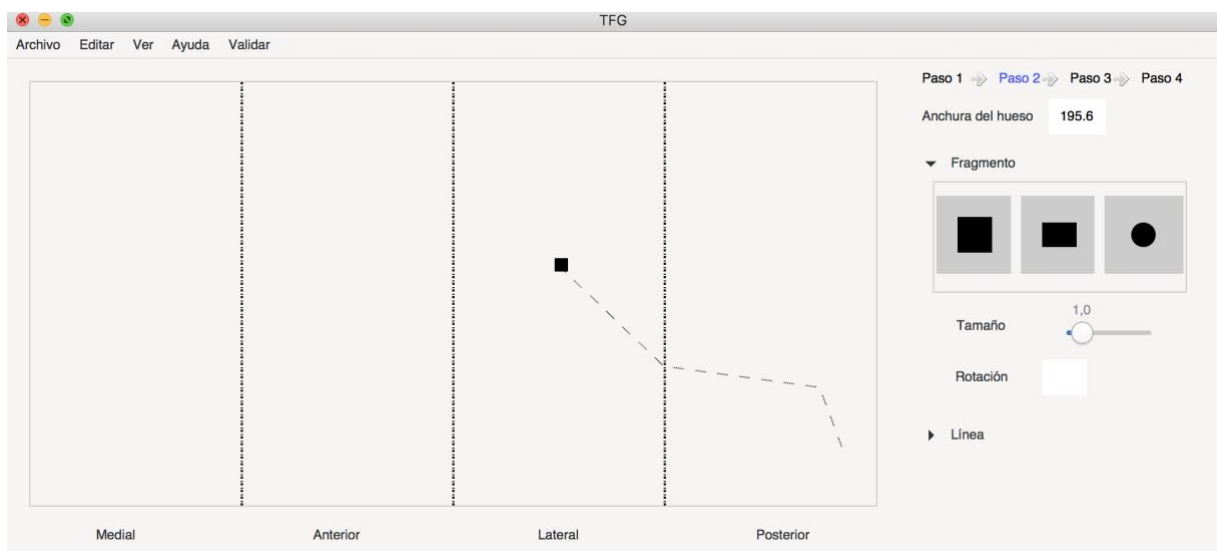


Ilustración 75. Operación rehecha

3.3. Pestaña “Archivo”. Nuevo

Permite la creación de un nuevo canvas en blanco a partir del cual comenzar a realizar la fractura desde 0.

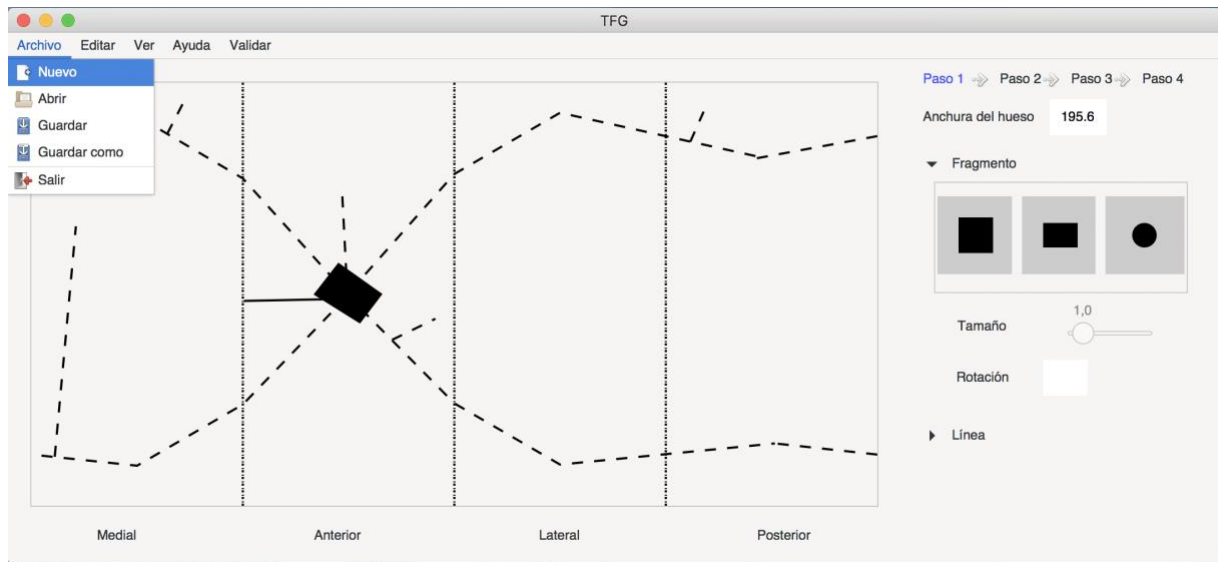


Ilustración 76. Crear un canvas nuevo

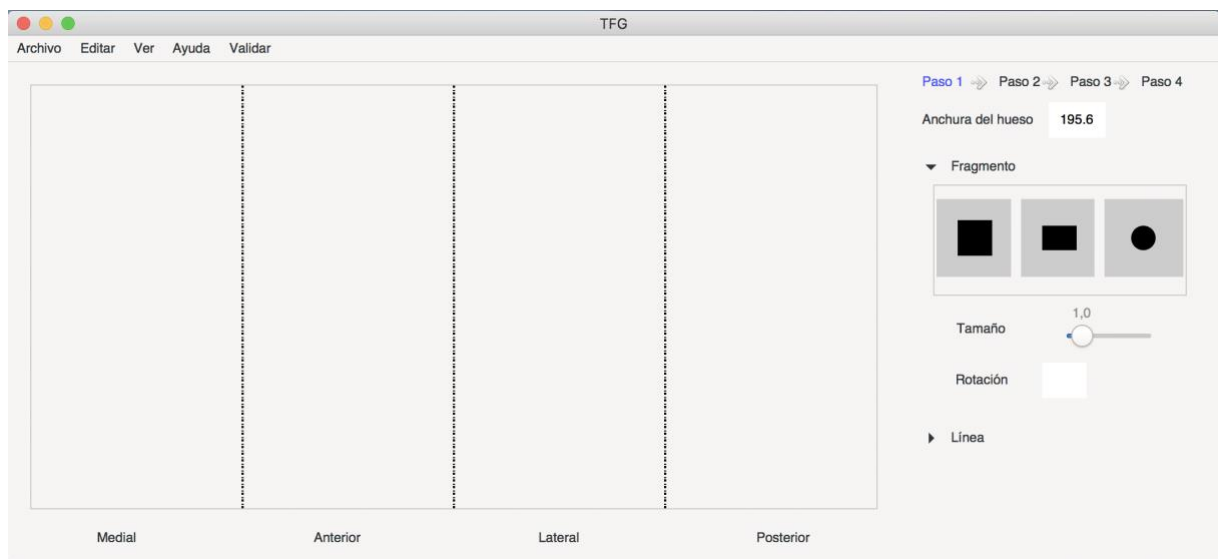


Ilustración 77. Nuevo canvas creado

3.4. Pestaña “Archivo”. Abrir

Permite cargar un archivo que contenga los datos de una fractura desde cualquier carpeta del equipo, previamente guardada desde este editor o desde otros, en nuestro canvas permitiendo su posterior modificación.

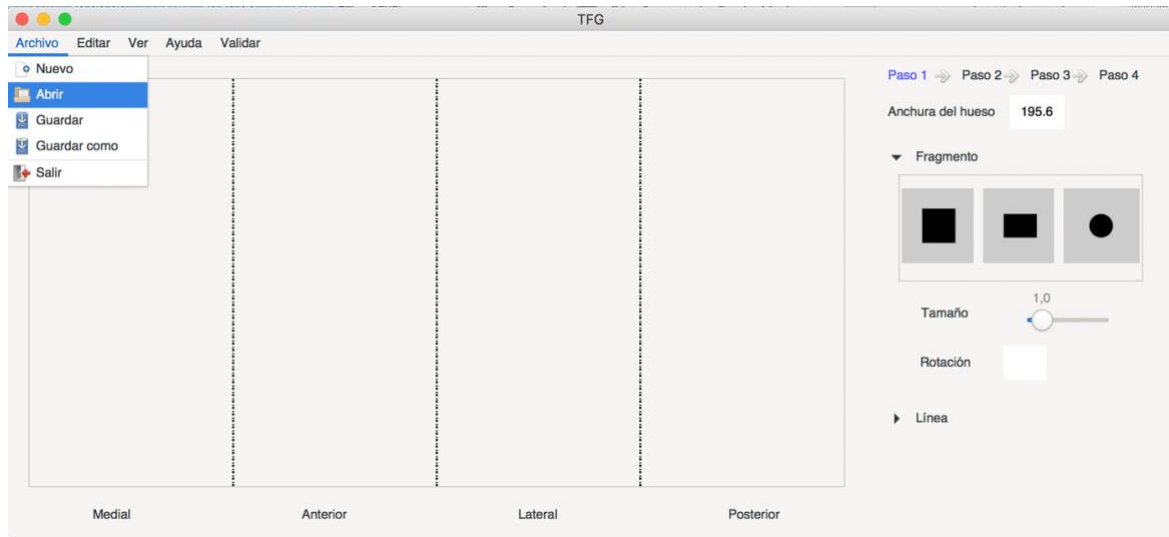


Ilustración 78. Cargar patrón de fractura

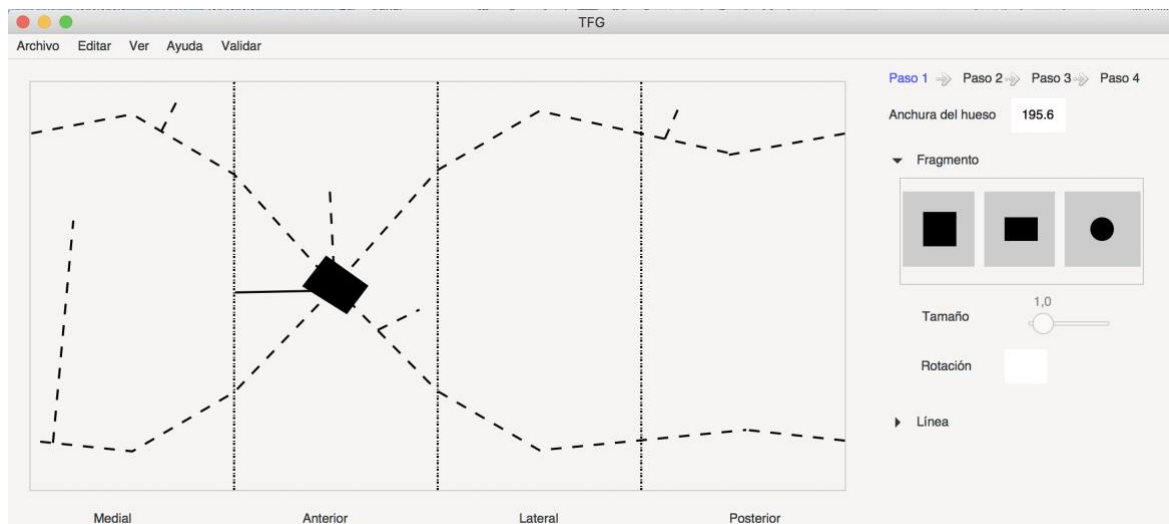


Ilustración 79. Patrón de fractura cargado

Una vez cargado el patrón de fractura se puede trabajar con él como si de un patrón realizado por la herramienta en ese momento se tratara, pudiendo deshacer el dibujado de sus líneas mediante el atajo de teclado indicado para ello o rehacer, rotar el fragmento, escalarlo, añadir nuevas líneas, validarlo, etc. En definitiva, la herramienta permite la edición completa del patrón de igual modo que si fuera un patrón de fractura hecho en “directo”.

3.5. Pestaña “Archivo”. Guardar

Esta función guarda los cambios en la fractura en caso de que se estuviera editando una fractura previamente cargada o guardada, o si es una fractura nueva, lo guarda con el nombre “undefined” con la extensión pf2d.

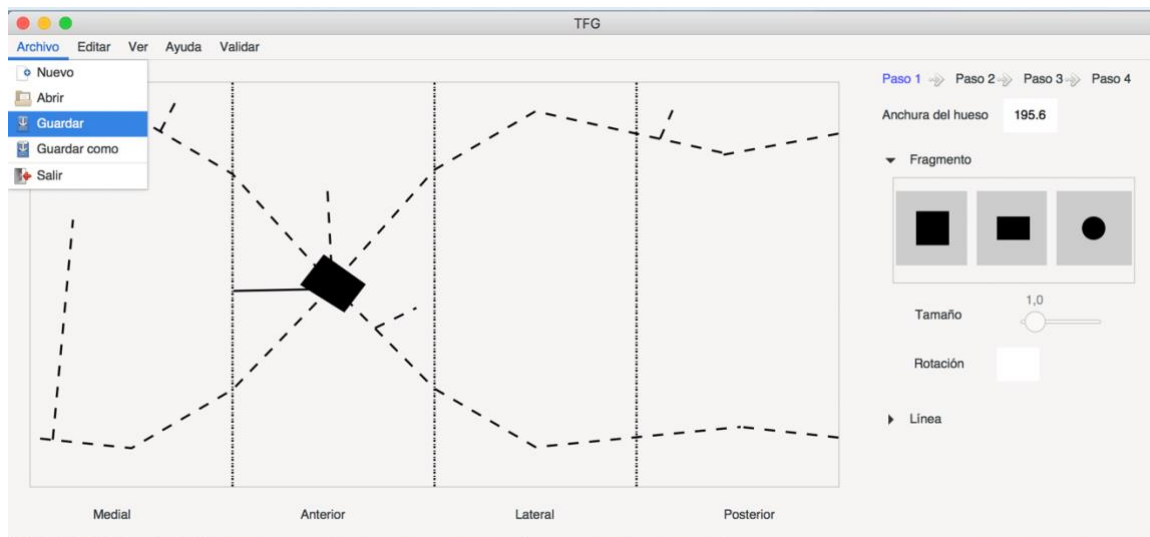


Ilustración 80. Guardado del patrón de fractura

3.6. Pestaña “Archivo”. Guardar como

Esta opción permite guardar una fractura controlando donde se quiere almacenar y qué nombre se le quiere dar al archivo.

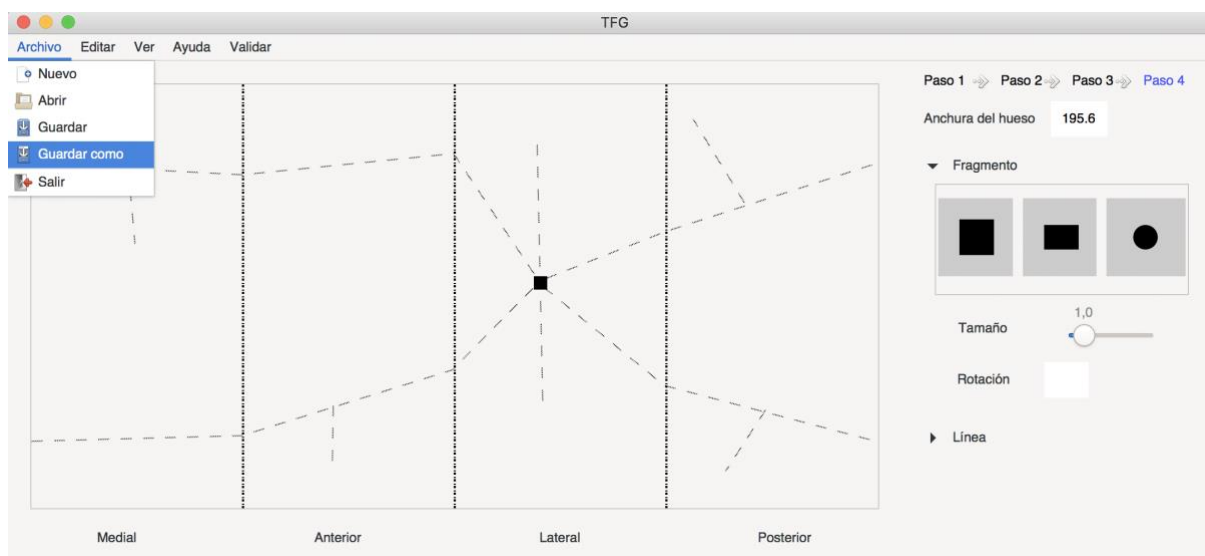


Ilustración 81. Guardar como

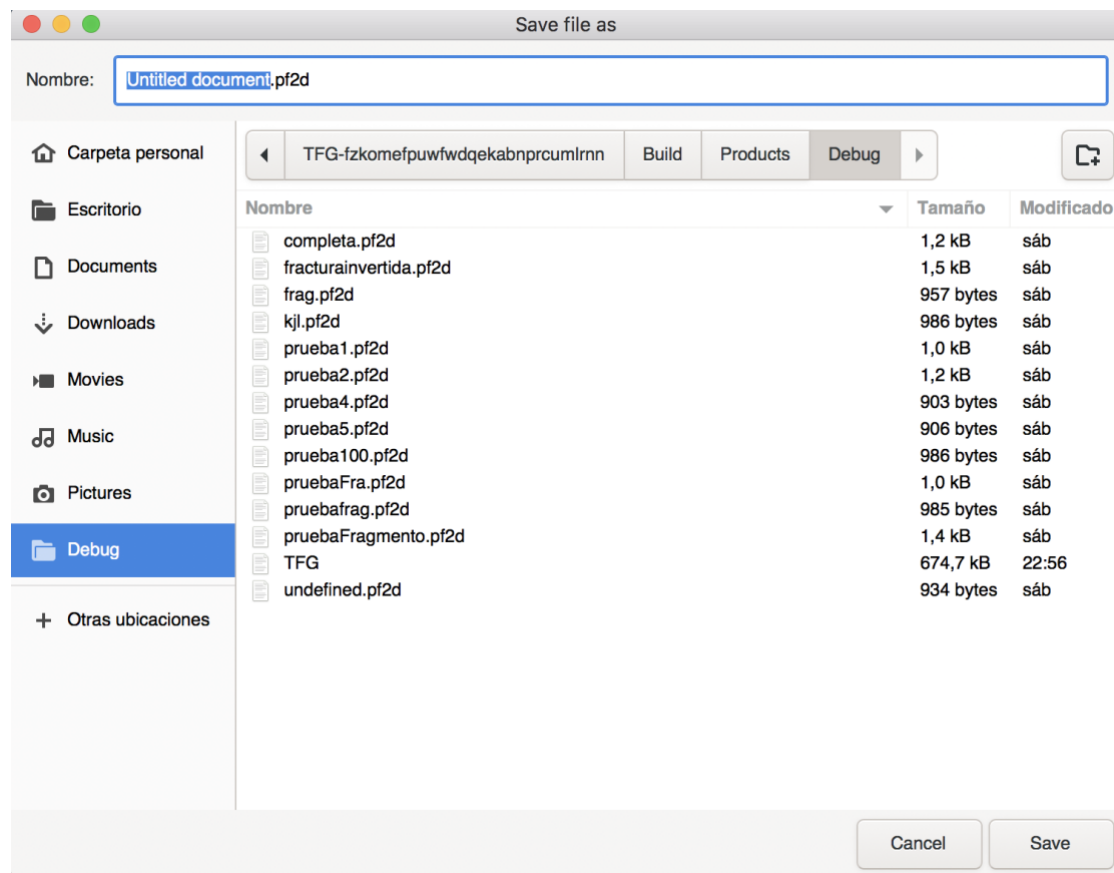


Ilustración 82. Ventana de guardado

3.7. Pestaña “Archivo”. Salir

Cierra el programa

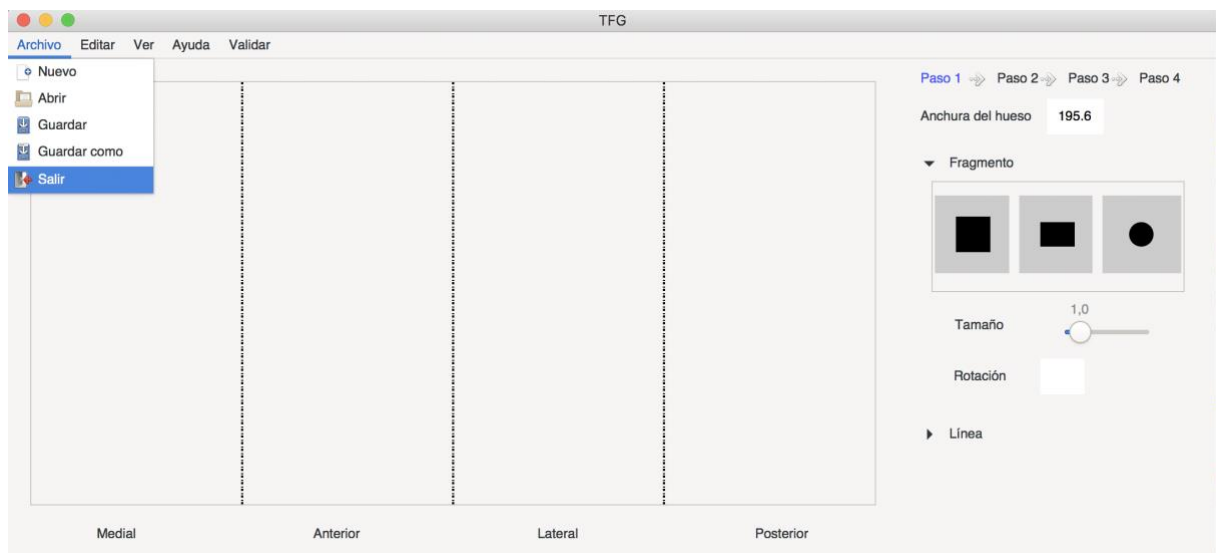


Ilustración 83. Salir del programa

3.8. Pestaña “Ayuda”

En esta pestaña tenemos un manual de usuario, el cual le explica de forma resumida al usuario los pasos a seguir para crear un patrón de fractura[.].

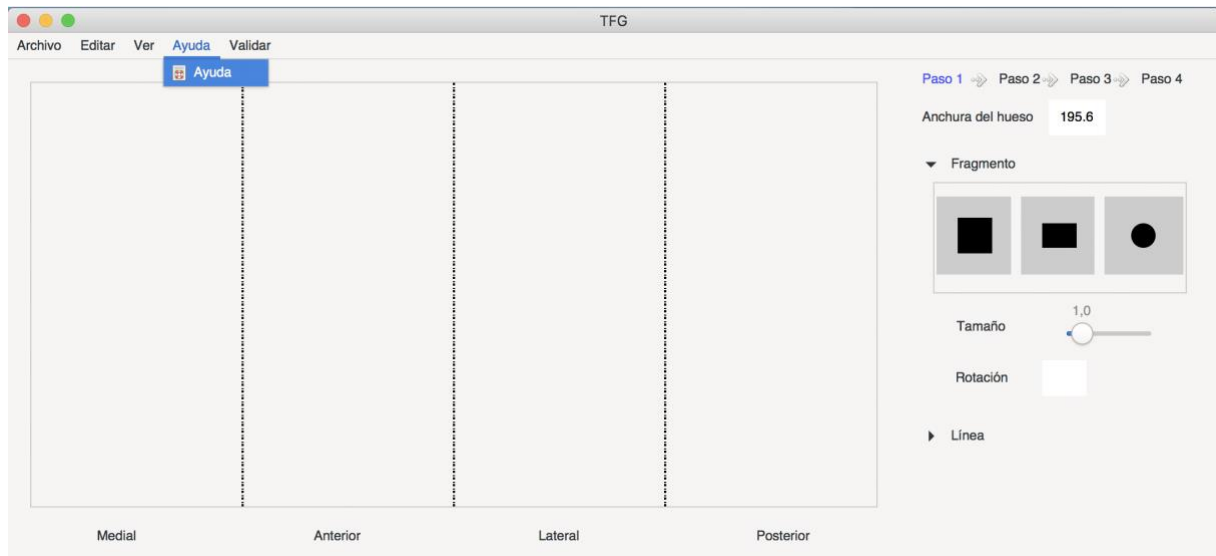


Ilustración 84. Ayuda

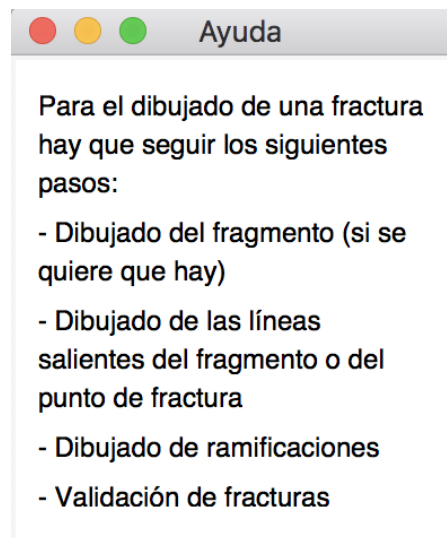


Ilustración 85. Ventana de ayuda

3.9. Pestaña “Ver”

En la pestaña ver encontramos los datos referentes a estudios de fracturas y estadísticas en función del tipo de fractura.

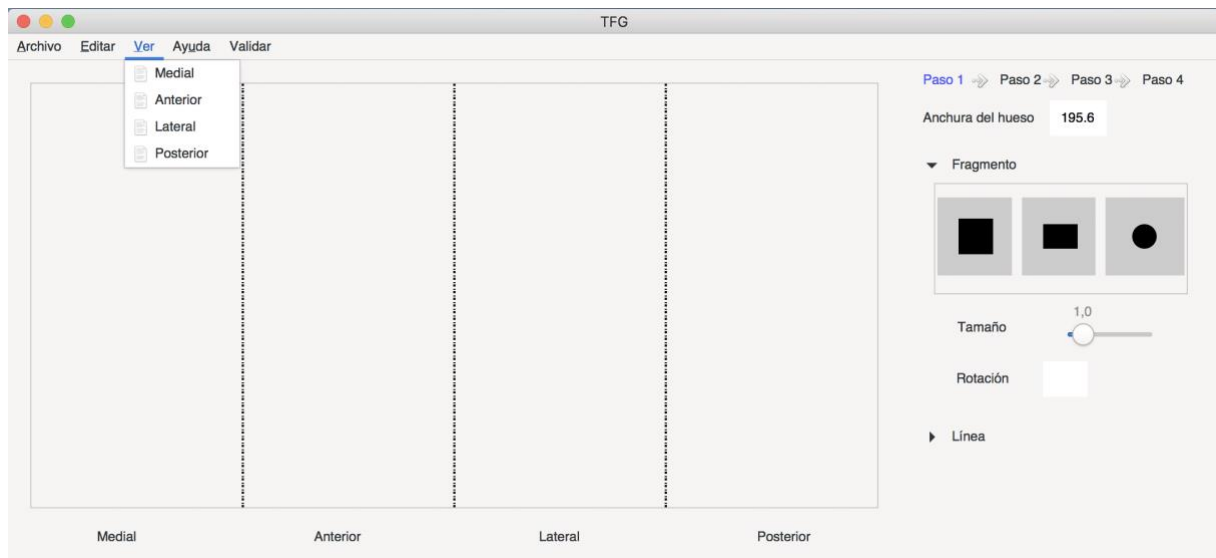


Ilustración 86. Pestaña VER

Características impacto lateral		
▼ Características según la dirección de impacto		
	Media	Desviación
Longitud de las líneas de fractura	3.6	0.9
Longitud de las líneas de fractura	176.6	71.5
Tamaño del fragmento	48.9	7.7

► Distribución de líneas longitudinales según la dirección de impacto
► Media de líneas de fractura según la dirección de impacto

Ilustración 87. Ejemplo pestaña ver lateral

3.10. Pestaña “Validar”

En esta pestaña se realiza la validación automática de la fractura creada. Tenemos tres tipos de test: de características, líneas longitudinales o líneas de fractura.

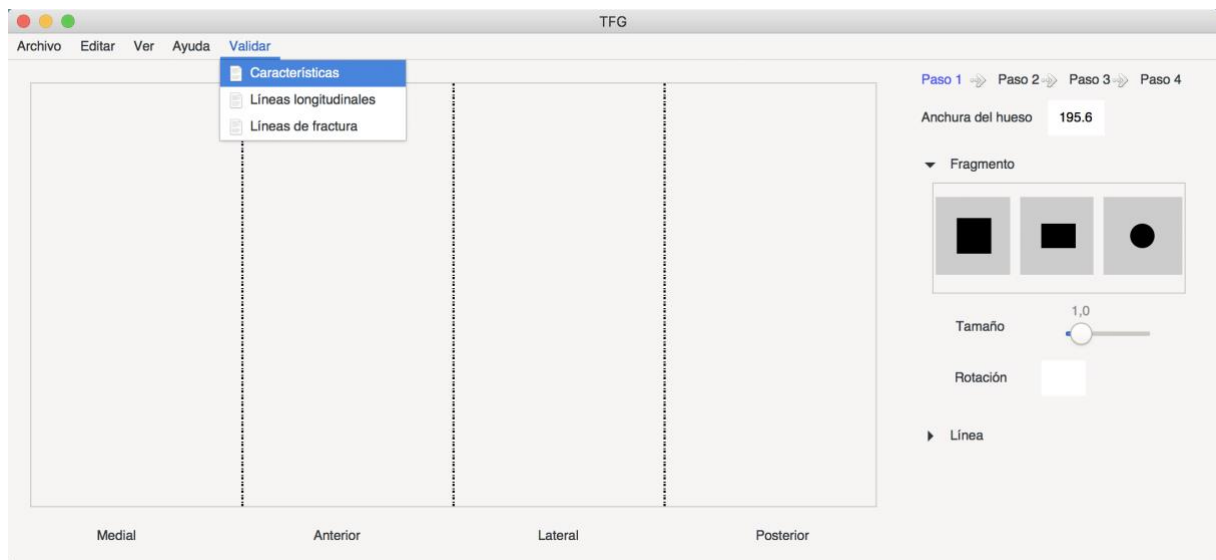


Ilustración 88. Pestaña Validar. Características

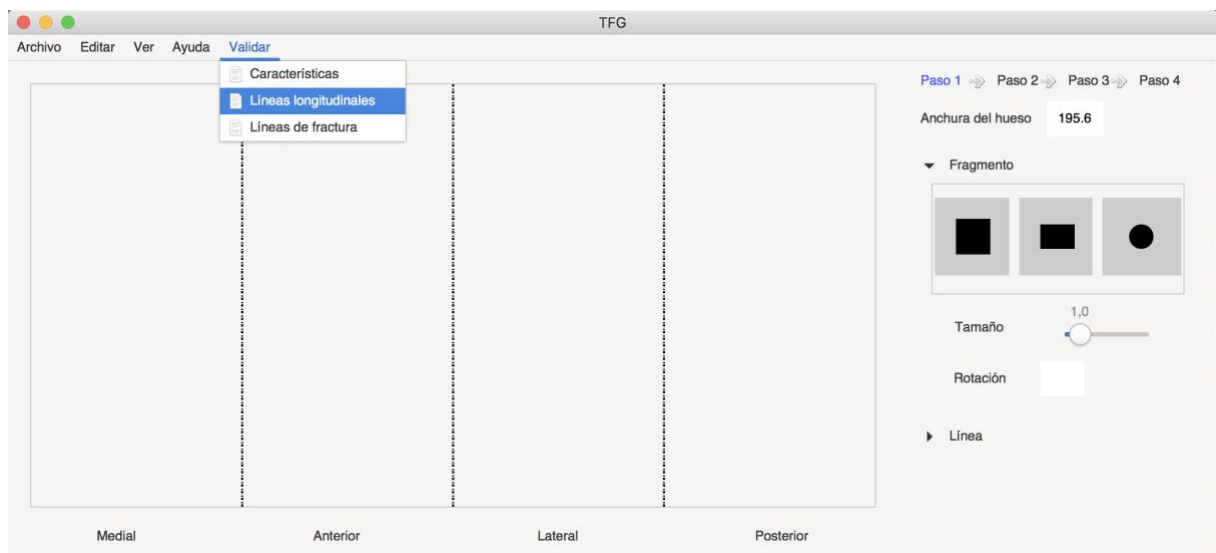


Ilustración 89. Pestaña validar. Líneas longitudinales

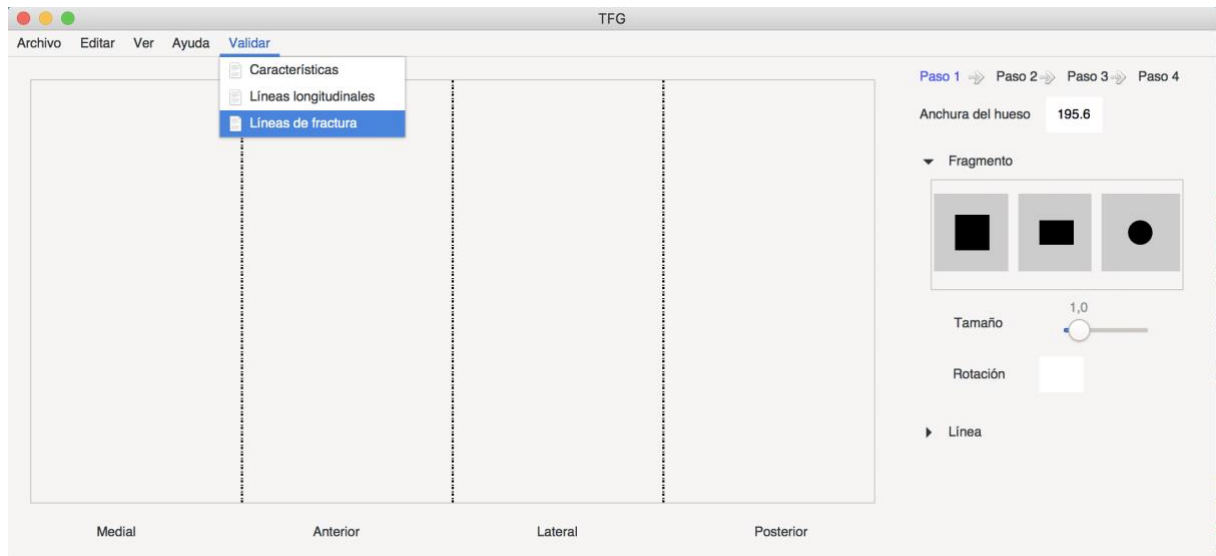


Ilustración 90. Pestaña validar. Líneas de fractura

Cada una de estas opciones de validación nos dan una valoración aproximada de corrección del patrón de fractura según los artículos científicos, siendo de mayor validez si se pasan todos los test de validación.

ANEXO II. Cuestionario de usabilidad**ASPECTOS DE USABILIDAD**

A continuación, se presentan una serie de cuestiones a responder para realizar un análisis de la herramienta previamente mostrada. Por favor, marque con una (X) su respuesta o indique la información requerida.

A- DATOS GENERALES

1. SEXO:

2. EDAD DEL USUARIO:

3. GRADO DE INSTRUCCIÓN

PRIMARIA ☐E.S.O ☐BACHILLERATO ☐SUPERIOR ☐TÉCNICA ☐UNIVERSITARIA ☐

A continuación, se procederá a evaluar el prototipo usado, para ello selecciones mediante un (X) el grado de conformidad del 1 al 5, siendo 1 el índice menos favorable.

B- DATOS ESPECÍFICOS

	1	2	3	4	5
1. Encuentro la herramienta fácil de usar					
2. Requerí de ayuda para realizar las tareas solicitadas					
3. Considero al producto fácil de usar para cualquier persona con conocimientos en el área y con capacidad de manejar una computadora o sistema					
4. Considero que no se requiere conocimiento previo para realizar las tareas solicitadas					
5. Mensajes e instrucciones ambiguos o incorrectos					
6. Número de pasos excesivos para realizar una tarea					
7. Considero fiable al producto software					
8. Considero seguro al producto software, en términos de fiabilidad					
9. Cometí uno/mucho error/errores durante la realización de las tareas, a causa de no leer claramente un mensaje o indicación del producto software.					
10. Cometí uno/muchos error/es durante la realización de las tareas a causa de que algún/os mensaje/s o indicación/es de la herramienta software no estaban claros					
11. El producto software tiene mensajes o instrucciones que no son claros para poder llevar a cabo una tarea					
12. Considero elegante o atractivo al producto software (combinación de colores adecuada, tamaño de letra legible, etc.)					
13. Estoy cómodo/a con el software					

Enlace para descargar el archivo:

https://drive.google.com/file/d/1nJzuwYN_bBohG_ZCjSmOYFHYyMuxEl6M/view?usp=sharing

ANEXO III. Cuestionario de validación de expertos**Validación de expertos**

A continuación, se presentan una serie de cuestiones a responder para realizar un análisis de la herramienta previamente mostrada. Por favor, marque con una (X) su respuesta o indique la información requerida.

A- DATOS DEL EXPERTO

1. EDAD:

2. EXPERIENCIA EN EL ÁREA

BAJA ☐MEDIA ☐ALTA ☐

A continuación, se procederá a evaluar el prototipo usado, para ello selecciones mediante un (X) el grado de conformidad del 1 al 5, siendo 1 el índice menos favorable.

B- CUESTIONARIO. TIPO TEST

	1	2	3	4	5
1. Encuentro la herramienta fácil de usar					
2. Requerí de ayuda para realizar las tareas solicitadas					
3. Considero al producto fácil de usar para cualquier persona con conocimientos en el área y con capacidad de manejar una computadora o sistema					
4. Mensajes e instrucciones ambiguos o incorrectos					
5. Número de pasos excesivos para realizar una tarea					
6. Considero seguro al producto software, en términos de fiabilidad					
7. Cometí uno/mucho error/errores durante la realización de las tareas, a causa de no leer claramente un mensaje o indicación del producto software.					
8. Cometí uno/muchos error/es durante la realización de las tareas a causa de que algún/os mensaje/s o indicación/es de la herramienta software no estaban claros					
9. El producto software tiene mensajes o instrucciones que no son claros para poder llevar a cabo una tarea					
10. Considero elegante o atractivo al producto software (combinación de colores adecuada, tamaño de letra legible, etc.)					
11. Estoy cómodo/a con el software					
12. El modo de representación del patrón de fractura una vez acabado se corresponde con el modo de representación habitual de patrones en esta área de investigación					
13. Los pasos a seguir para realizar el dibujado de un patrón son intuitivos					
14. Los pasos necesarios para crear un fragmento son intuitivos					

15. Los pasos necesarios para modificar un fragmento son sencillos					
16. Los pasos necesarios para ajustar las características de una línea de fractura son sencillos					
17. La secuencia de pasos para dibujar una línea oblicua son los necesarios					
18. En general, el dibujado de las líneas de fractura es intuitivo y cómodo de realizar					
19. He necesitado la pestaña de ayuda de la herramienta para saber comenzar a usar la herramienta					
20. He necesitado la pestaña ayuda mientras usaba la herramienta					
21. Grado de satisfacción con la herramienta					
22. Volveré a usar la herramienta si tengo que hacer una tarea relacionada con lo ofrecido en el prototipo					

A continuación, se presentan una serie de cuestiones a responder para realizar un análisis de la herramienta previamente mostrada. Por favor, rodee con un círculo su respuesta o indique la información requerida.

C- CUESTIONARIO. RESPUESTA ABIERTA

1. En cuanto a la validación de la fractura:

- El sistema usado ha valorado aspectos en el patrón como correcto y bajo mi experiencia la fractura es correcta.
- El sistema usado a valorado los aspectos en el patrón como correcto y bajo mi experiencia en el área la fractura es incorrecta.
- Es sistema usado ha valorado aspectos en el patrón como incorrectos y bajo mi experiencia la fractura es correcta.
- El sistema usado ha valorado aspectos en el patrón como incorrectos y bajo mi experiencia la fractura es incorrecta.

En caso de no haber coincidido con el resultado dado por el validador de la herramienta, justifique brevemente por qué no lo es:

.....

2. ¿Cree que debería ser más guiada la generación del patrón?

- Sí
- No

Si la respuesta es afirmativa, ¿por qué?

.....

3. Sugerencias

.....

En el siguiente enlace se puede descargar el documento:

https://drive.google.com/file/d/1qIZVWQuN_Glc9IVrraFrEaASHPWwfmYQ/view?usp=sharing

ANEXO IV. Creación de un ejecutable en xCode e instalación en otros equipos

Para la creación de un ejecutable que se pueda exportar a otros equipos y se pueda usar a la perfección se han seguido una serie de pasos:

Primero vamos a crear un esquema nuevo para release:

4. Ir al proyecto en el xCode, y en el esquema activo crear un nuevo esquema.

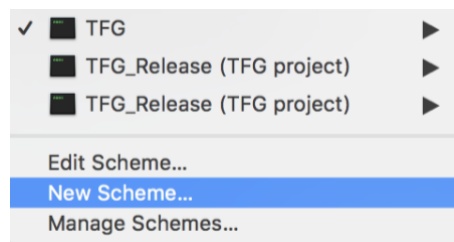


Ilustración 91. Creación de un esquema

5. Una vez creado dicho esquema, cambiarlo de debug a release. Para ello pulsar en Edit Scheme y en la pestaña Run → Info, cambiar la opción de debug a release.
6. Cuando ya tenemos preparado el nuevo esquema, vamos a la pestaña target y pulsamos sobre nuestro proyecto. Una vez hecho esto buscamos la pestaña Build Phases y en Copy Files, seleccionamos configuramos Destination → Absolute Path y dejamos el path en blanco.

Seleccionamos Copy only when installing y añadimos mediante el simbolo más la carpeta opt, lib y Cellar, previamente creada a partir de los paquetes que tenemos en dichas carpetas en nuestro sistema pero únicamente con los paquetes necesarios para el funcionamiento del programa.

7. Ir a la pestaña Product → Archive y empaquetar el ejecutable como Build Products. Una vez hecho esto, ya se puede distribuir el resultado.

Una vez llegado a este punto, encontramos una carpeta con el ejecutable y otras carpetas. Para el correcto funcionamiento en otro ordenador necesitamos hacer la instalación.

8. En la carpeta /usr/local/opt, introducir el contenido de opt
9. En la carpeta /usr/local/Cellar introducir el contenido de Cellar
10. En la carpeta /usr/local/lib introducir el contenido de la carpeta lib
11. Depositar el ejecutable del proyecto en la carpeta /usr/local/bin
12. Finalmente, introducir en Applications nuestro proyecto.