



Facultad  
de  
Ciencias

**AUTOMATIZACIÓN DEL ANÁLISIS DOSIMÉTRICO Y  
VERIFICACIÓN DE LOS PLANES DE TRATAMIENTO  
DE RADIOTERAPIA EXTERNA EN EL HOSPITAL  
UNIVERSITARIO MARQUÉS DE VALDECILLA**

(Automatization of the dosimetric analysis and the external  
radiotherapy treatment plans at the local hospital)

Trabajo de Fin de Grado  
para acceder al

**GRADO EN FÍSICA**

**Autor:** Nicolás Gillard Barroso

**Director:** Saturnino Marcos Marcos

**Co-directora:** Marina Gutiérrez Ruiz

**Mayo 2023**

## Resumen

En radioterapia, el análisis de los histogramas dosis-volumen (HDV) de los órganos de riesgo y los volúmenes blanco es una estrategia sencilla y eficaz pero resulta en un trabajo laborioso cuando el número de órganos a comprobar es elevado (cáncer de cabeza y cuello o tratamientos de SBRT). Además, durante la revisión del plan, hay que verificar numerosos parámetros de la planificación (origen, mesa de tratamiento, algoritmo de cálculo...) que supone una labor repetitiva que consume un tiempo significativo del proceso radioterápico y que está sujeta a potenciales errores humanos. Para ahorrar tiempo y minimizar posibles errores, se ha diseñado el *script* "Check-list verificación tratamientos SOR", que realiza de forma automática el análisis dosimétrico y la verificación de los planes de tratamiento de radioterapia externa del Servicio de Oncología Radioterápica (SOR) del Hospital Universitario Marqués de Valdecilla (HUMV). Se ha analizado la eficacia del *script*, obteniendo que para tumores de cabeza y cuello se tiene un ahorro de tiempo del 50 %, aproximadamente.

**Palabras clave:** Radioterapia, histogramas dosis-volumen, check-list, tiempo, errores humanos, *script*.

## Abstract

In radiotherapy, the analysis of the dose-volume histograms (DVH) of the organs at risk and target volumes is a simple and effective strategy but can result in a laborious task when the number of organs to check is large (head and neck cancer or SBRT treatments). Also, during the plan revision, many planification parameters (origin, treatment couch, calculation algorithm...) have to be checked, which involves a repetitive work that consumes significant time in the radiotherapy process and is subject to potential human errors. To save time and minimize possible errors, the script "Check-list verificación tratamientos SOR", which automatizes the dosimetric analysis and the treatment plans check of the Radiotherapy Oncology Service (SOR) of Hospital Universitario Marqués Valdecilla (HUMV) has been created. A study in which the effectiveness of this script has been tested has also been made, obtaining approximately a 50 % time save for tumors located in the head and neck.

**Key words:** Radiotherapy, dose-volume histograms, check-list, time, human errors, script.

# Agradecimientos

Lo primero de todo, agradecer a mis tutores en este trabajo, Saturnino Marcos Marcos y Marina Gutiérrez Ruiz, que han sido muy pacientes y se han involucrado mucho en la creación del mismo.

Me gustaría dejar claro mi agradecimiento a Alejandro Barranco López, del Hospital Clínico Universitario Lozano Blesa, que tras contactar con él se ha volcado en el trabajo y ha ayudado en todo lo posible.

En el ámbito más personal, me gustaría agradecer a todo aquel amigo o amiga que me haya apoyado de alguna manera u otra.

Dedico todo a mi padre David John Gillard, a mi madre M<sup>a</sup> Ángeles Barroso y a mis abuelos Enrique Barroso y María Villa, quienes en todo momento me han mostrado su interés y apoyo. John Gillard you will be forever remembered. R. I. P.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Radiación ionizante . . . . .	2
1.1.1. Radiación electromagnética . . . . .	2
1.1.2. Radiación corpuscular . . . . .	3
1.1.3. Dosis absorbida en función de la profundidad de la radiación en un medio material . . . . .	4
1.2. Interacción de la radiación con el medio celular . . . . .	5
1.2.1. Formas de interacción de la radiación electromagnética con el medio celular . . . . .	5
1.2.2. Daño radioinducido en el ADN . . . . .	6
1.3. La radioterapia . . . . .	7
1.3.1. Fundamentos de la radioterapia . . . . .	7
1.3.2. Linear Accelerators (LINACs) . . . . .	8
1.3.3. Etapas del proceso radioterápico . . . . .	11
<b>2. Hipótesis y objetivos</b>	<b>14</b>
<b>3. Material y método experimental</b>	<b>15</b>
3.1. Red ARIA . . . . .	15
3.2. Planificador Eclipse . . . . .	15
3.3. Eclipse Scripting API . . . . .	16
3.4. <i>Check-list</i> de verificación de los tratamientos . . . . .	17
3.5. Diseño del <i>script</i> . . . . .	21
3.6. Pacientes para comprobar el código . . . . .	22
3.7. Ahorro de tiempo en la etapa de verificación del tratamiento . . . . .	28
<b>4. Resultados y análisis</b>	<b>30</b>
4.1. Funciones del código . . . . .	30
4.2. Ejecución del código . . . . .	35
4.3. Ahorro de tiempo en la etapa de verificación del tratamiento . . . . .	39
<b>5. Discusión</b>	<b>40</b>
<b>6. Conclusiones y trabajo a futuro</b>	<b>44</b>

Bibliografía	45
Anexo I: Código completo	48
Anexo II: Estimación de errores	73

# Capítulo 1

## Introducción

El aumento de la esperanza de vida debido a la eliminación de enfermedades que causan una muerte temprana, hace que la incidencia del cáncer aumente. Esto último ha hecho que el cáncer, como enfermedad potencialmente curable, tenga una repercusión social y económica universalmente aceptada. En el año 2022, concretamente, aparecieron 290.175 casos nuevos de cáncer según cifras de la Asociación Española contra el Cáncer, lo cual supuso un aumento del 1,34% respecto al año anterior [1].

La radioterapia, como herramienta para tratarlo, se utiliza desde principios del siglo veinte [2], y ha evolucionado con los avances científicos de la física y de la oncología. Actualmente es una de las modalidades más efectivas de tratamiento para el cáncer, junto con la cirugía y la quimioterapia.

La radioterapia es una forma de tratamiento basada en el empleo de radiaciones ionizantes (rayos X, rayos gamma, electrones o protones) para eliminar o dañar las células cancerígenas, las cuales crecen y se dividen más rápidamente que las células normales. La radiación actúa sobre la cadena de ADN que se encuentra dentro de las células produciendo pequeñas roturas, las cuales evitan que las células cancerígenas crezcan, se dividan y finalmente mueran.

El conocimiento general acerca de la radioterapia, sin embargo, no suele ir más allá de esta idea. Se sabe que determinadas acciones como tomar el sol sin crema o fumar propician la aparición del cáncer, pero se desconoce la biofísica del proceso. Por ello, en los próximos apartados definiremos el concepto de radiación ionizante así como los tipos de radiación existentes y sus procesos de interacción con los átomos del medio. A continuación, se describe cómo la radiación interacciona con el medio celular y afecta en particular a la molécula de ADN dando lugar a un daño radioinducido. Finalmente, hablaremos de la utilidad médica que recoge y aplica todo lo anterior, la Radioterapia.

## 1.1. Radiación ionizante

La radiación ionizante consiste en partículas (alfa, beta, protones, neutrones...) y radiación electromagnética (fotones de rayos X, rayos gamma...) capaces de ionizar los átomos. La ionización es el proceso de arrancar uno o más electrones de los átomos al absorber éstos energía de la radiación incidente, dejando atrás partículas cargadas eléctricamente (generalmente un electrón y un ión cargado positivamente).

La radiación ionizante se encuentra en todas partes, como puede ser en forma de rayos cósmicos provenientes del espacio exterior o de isótopos radiactivos originados naturalmente como el potasio, entre otros. En función de la exposición a esta radiación, pueden aparecer efectos biológicos perjudiciales o no. La exposición a dosis muy altas de radiación ionizante puede provocar daños inmediatos en el cuerpo, que incluye daños graves en la piel o los tejidos, enfermedad aguda por radiación y muerte [3]. La radiación ionizante, sin embargo, puede también tener aplicaciones beneficiosas, como puede ser el uso de rayos gamma, haces de electrones, protones o iones para el tratamiento de diferentes tipos de cáncer.

Las partículas cargadas, incluso con poca velocidad, son directamente ionizantes, es decir, tienen suficiente energía como para liberar electrones de los átomos o moléculas con los que interactúan, creando así iones cargados eléctricamente [4]. La radiación electromagnética (EM) puede producir ionización directa (mediante efecto fotoeléctrico, Compton o producción de pares) y además indirecta mediante la ionización que producen las partículas cargadas resultantes de la interacción directa.

### 1.1.1. Radiación electromagnética

La radiación electromagnética tiene naturaleza corpuscular y ondulatoria. Dentro del espectro de radiación electromagnética se encuentran, de menor a mayor energía, las ondas de radio, las microondas, los infrarrojos, la luz visible, la luz ultravioleta, los rayos X y los rayos Gamma.

En el día a día el ser humano está expuesto a muchas de estas radiaciones (microondas, infrarrojos, visible y ultravioleta de baja energía), pero por lo general, no resultan un peligro, pues éstas no son ionizantes. Sin embargo, la exposición a luz ultravioleta de alta energía, rayos X o rayos Gamma puede suponer un grave peligro por tratarse de radiación ionizante.

Los fotones pueden interactuar con la materia según su energía y el número atómico del medio en el que indican a través de los siguientes mecanismos [5]:

- **Absorción fotoeléctrica:** ocurre cuando el fotón interactúa con un electrón ligado del átomo y le transfiere toda su energía. Si el fotón tiene más energía que la energía de ligadura del electrón, éste es arrancado de su capa mientras que el fotón es absorbido en el proceso. Este efecto es el más probable para fotones de baja energía y electrones más ligados.
- **Efecto Compton:** ocurre cuando el fotón incidente interactúa con un electrón de un átomo del medio, cuya energía de enlace es mucho menor que la del fotón. El fotón cede

al electrón parte de su energía, provocando su expulsión del átomo, y se dispersa con la energía restante, cambiando su dirección respecto a su dirección incidente.

Se considera que es predominante para fotones con energías en el rango de kiloelectronvoltios (keV) a varios megaelectronvoltios (MeV). Los aceleradores utilizados en radioterapia trabajan en el orden de los megaelectronvoltios, por lo que este efecto es el más relevante para la radiación electromagnética que generan.

- Producción de pares:** cuando un fotón tiene una energía superior a 1,02 MeV, es decir, dos veces la energía en reposo del electrón (0,511 MeV), puede darse este proceso en el cual el fotón se materializa en un electrón y un positrón (la antipartícula del electrón). Este efecto es el más probable para fotones de alta energía, comenzando a ser relevante a partir de 2 MeV de energía.

A continuación, se muestra un diagrama en el que se puede ver con más detalle qué efecto es el dominante en función de la energía del fotón y el número atómico de los átomos del medio, es decir, el número de protones en los núcleos de los átomos.

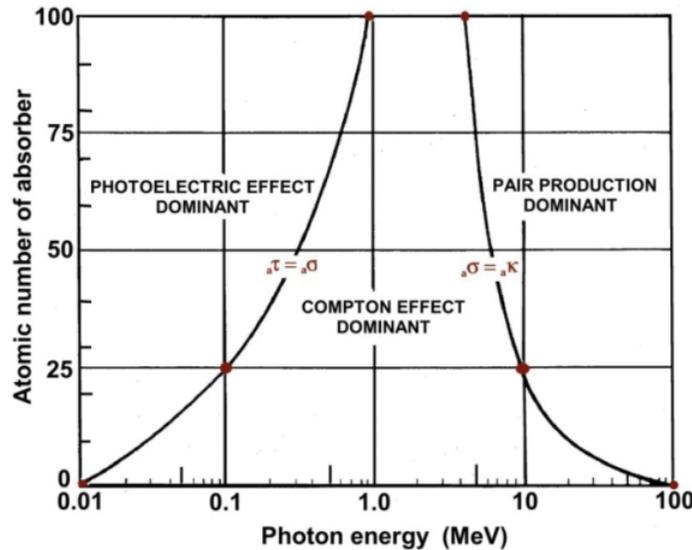


Figura 1.1: Interacciones dominantes para radiación electromagnética en función del número atómico del medio,  $Z$ , y la energía del fotón de la radiación incidente,  $E\gamma$ . Fuente: Kaunas University of Technology [6].

### 1.1.2. Radiación corpuscular

La radiación corpuscular se refiere a partículas subatómicas, como los electrones, protones y las partículas alfa, cuya energía o velocidad son suficientemente elevadas para poder ionizar los átomos o moléculas..

Las partículas cargadas pierden su energía en el medio al interactuar eléctricamente con las cargas de los electrones y protones de los átomos del mismo. Estas interacciones pueden dar

lugar a tres tipos de colisiones o procesos: [5]:

- **Colisión elástica:** La partícula choca con los átomos del medio, desviándose de su trayectoria y cediendo una cierta cantidad de energía en forma de energía cinética.
- **Colisión inelástica:** La partícula interacciona con los electrones atómicos transfiriendo a estos pequeñas cantidades de energía. La energía transferida puede provocar que algún electrón atómico escape de la atracción del núcleo produciendo la ionización del átomo, o que algún electrón atómico pase a un estado menos ligado produciéndose en este caso la excitación del átomo.
- **Colisión radiativa:** La partícula cargada se “frena” o se “desvía” en su interacción con los átomos del medio y como resultado emite ondas electromagnéticas. Esta “radiación” se conoce como radiación de frenado o Bremsstrahlung.

### 1.1.3. Dosis absorbida en función de la profundidad de la radiación en un medio material

La dosis se define como la cantidad de energía absorbida por unidad de masa cuando una persona o un objeto es expuesto a radiación [7]. Los diferentes procesos de interacción de la radiación con la materia dan lugar a diferentes formas de deposición de la energía en función de la profundidad que alcanza la radiación en el medio. El cómo deposita la energía una determinada partícula en el medio material puede ser de gran utilidad en determinados campos como puede ser el de la radioterapia. En función de dónde esté localizado un tumor, conviene elegir un tipo de radiación u otro.

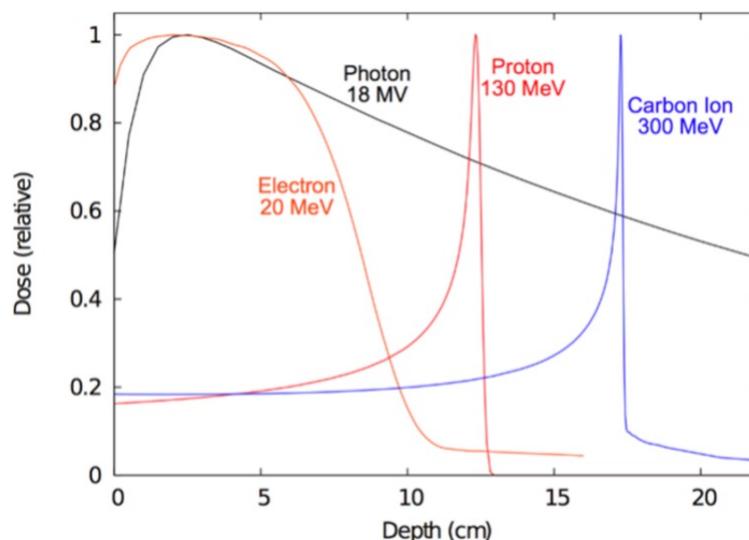


Figura 1.2: Dosis relativa recibida en función de la profundidad en el medio para fotones, electrones, protones e iones de carbono de altas energías. Fuente: Sociedad Española de Física Médica [8].

Los fotones de entre 6 y 18 MeV presentan un máximo de deposición de energía de entre 1,5 y 3 cm de profundidad (esta propiedad se debe a los efectos de la radiación secundaria). Tras el máximo, su caída es suave, por lo que tienen una alta capacidad de penetración en el medio. Los electrones, sin embargo, tienen un rango corto, es decir, depositan su máximo de energía próximo a la superficie del material y tienen una caída de dosis abrupta. La dosis es prácticamente nula más allá de unos pocos centímetros de profundidad, por lo que en radioterapia son de especial interés para tratar tumores superficiales, como es el caso de la piel o los ojos. En el caso de los protones, estos apenas depositan energía a la entrada del material, sino que atraviesan una cierta profundidad hasta que al final de su recorrido son frenados por completo, siendo ahí donde depositan su energía localmente. Esto se debe a la relación de masas entre los protones y los electrones. Cuando los protones o electrones tienen mucha energía y, por ende, mucha velocidad, están cerca de un determinado electrón durante un tiempo muy corto en comparación con cuando son más lentos y le transfieren un momento menor. La masa de los protones es aproximadamente mil veces superior a la de los electrones, por lo que apenas se ve afectada su dirección, mientras que los electrones, al tener muy poca masa, cambian de dirección con facilidad, emitiendo radiación de frenado y perdiendo energía. Eventualmente un protón puede colisionar con el núcleo de un átomo, cediéndole toda su energía. La ventaja del perfil de deposición de energía de los protones desde el punto de vista terapéutico es que si el tumor se encuentra a una determinada profundidad, se puede incidir con protones de una energía tal que dicho tumor se encuentre en este pico de máxima deposición, minimizando la energía depositada en el tejido sano que lo rodea.

## 1.2. Interacción de la radiación con el medio celular

Puesto que ya se ha hecho una introducción acerca de la radiación y sus tipos, ahora se pretende abordar la interacción de esta radiación con la materia, específicamente, sobre la materia viva.

### 1.2.1. Formas de interacción de la radiación electromagnética con el medio celular

La acción de la radiación ionizante sobre las células viene determinada por procesos de excitación e ionización, ya sea en el material genético (ADN), en cuyo caso se hablaría de acción directa, o en el citoplasma (principalmente en el agua), denominado acción indirecta [9].

I **Acción Directa:** La acción directa, también llamada “Teoría del impacto” , tiene lugar cuando un fotón ioniza o excita átomos de un blanco biológico (ADN, ARN, etc), afectando al funcionamiento de la célula.

II **Acción Indirecta:** La acción indirecta, también llamada "Teoría de los Radicales Libres", implica la absorción de la energía disipada en medios intracelulares, principalmente agua. Esta absorción puede dar lugar a la formación de radicales libres, que son átomos o moléculas que contienen un electrón de un orbital externo desapareado. Los radicales libres presentan una alta reactividad química por la tendencia del electrón libre a combinarse con otro electrón de un átomo de una molécula contigua [10].

En el caso del agua, que es el más común, generalmente se crea el radical libre hidroxilo  $\text{OH}^*$ , que se produce cuando se rompe un enlace del agua y queda un átomo de oxígeno con carga negativa ( $\text{O}^{*-}$ ) y un hidrógeno con carga positiva ( $\text{H}^+$ ).

De acuerdo con varios estudios [12][13], el daño indirecto a través de la producción de especies reactivas de oxígeno es más común que el daño directo en la cadena de ADN. Esto se debe a que las especies reactivas de oxígeno pueden viajar a través del cuerpo y afectar a las células a cierta distancia de donde se produjeron originalmente, mientras que el daño directo se limita al área donde ocurrió la radiación. Por esto, se va a dar más énfasis en el daño indirecto.

Los iones originados por la radiación tienden preferentemente a recombinarse, formando agua, en cuyo caso no se producen daños importantes. Sin embargo, debido a su alta reactividad, los radicales libres pueden tomar parte en otras reacciones como pueden ser la formación de otras moléculas o la formación de nuevos radicales libres en sustancias próximas, generándose una reacción en cadena. Si en el proceso se produce daño sobre moléculas no fundamentales, como pueden ser los carbohidratos, lípidos o proteínas, aunque sean alterados sólo se producen efectos transitorios, ya que inmediatamente se sintetizan los constituyentes dañados en el metabolismo normal, no produciéndose efectos apreciables [14]. Sin embargo, si la radiación llega a alcanzar blancos como, por ejemplo, la cadena de ADN, la célula puede sufrir mutaciones. Una mutación, por definición, es cualquier cambio en la secuencia del ADN [15]. Generalmente, las mutaciones se producen de forma natural mediante errores en la replicación del ADN [16].

### 1.2.2. Daño radioinducido en el ADN

La estructura del ADN se compone de dos cadenas complementarias que forman una doble hélice. Cada cadena está compuesta por una secuencia de nucleótidos, los cuales, a su vez, están formados por una base nitrogenada (adenina, guanina, citosina o timina). Las bases nitrogenadas de ambas cadenas están unidas mediante enlaces de hidrógeno. La información genética, es decir, las instrucciones y regulaciones de los procesos celulares, está codificada en la secuencia de nucleótidos de cada cadena [14].

Los problemas que pueden ocurrir en el ADN, ya sea por acción de los radicales libres o por acción directa, pueden ser la sustitución o pérdida de una base nitrogenada, la ruptura del enlace de hidrógeno o la ruptura de la estructura en una o ambas cadenas, entre otras, dando origen a una mutación.

Las células disponen de mecanismos de detección y reparación como puede ser la unión de

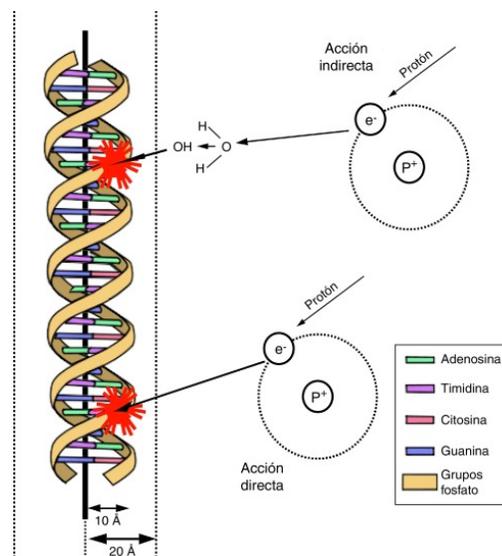


Figura 1.3: Esquema ilustrativo de las acciones directa e indirecta de la radiación. Fuente: ScienceDirect [11].

los extremos a ambos lados de la rotura. En este proceso, que es el más frecuente, el código genético es eliminado. Si el código genético perdido es esencial, las consecuencias pueden ser graves, estando entre ellas la aparición del cáncer. El cáncer es una enfermedad causada por una mutación en los genes que regulan el ciclo de la célula. Como consecuencia, las células se reproducen fuera de control, formando masas anormales y, a diferencia de las células sanas, estas no se autodestruyen cuando dejan de funcionar correctamente, por lo que la muerte celular sólo puede producirse por un factor externo, como puede ser la radioterapia.

Este descontrol reproductivo de células es un problema porque las células cancerosas compiten con las sanas por los nutrientes y el oxígeno, y porque forman tumores que comprimen los tejidos y órganos cercanos. Si se alcanza la fase de metástasis, las células cancerosas pueden propagarse a otras partes del cuerpo a través del sistema circulatorio o linfático, causando daños a órganos y tejidos lejanos y provocando potencialmente la muerte del individuo. En el 80% de las muertes por cáncer se ha alcanzado la metástasis, por lo que es muy importante detectarlo y tratarlo lo antes posible. (Más información acerca del cáncer en [17]).

## 1.3. La radioterapia

A día de hoy existen tres métodos fundamentales [18] para el tratamiento del cáncer: la cirugía, donde el tumor es extirpado, la quimioterapia, donde las células cancerosas mueren por el uso de fármacos, o la radioterapia, donde se utiliza radiación ionizante de alta energía para destruir células cancerosas.

### 1.3.1. Fundamentos de la radioterapia

El hecho de que las células cancerosas se dividan y crezcan de forma descontrolada las hace más susceptibles a la radiación ionizante. Aprovechando esto, en la radioterapia se utiliza radiación ionizante que penetra en los tejidos y afecta al ADN de las células cancerosas, provocando daños en su material genético e impidiendo su capacidad para crecer y reproducirse, provocando su muerte.

Como se vio en la Figura 1.2, no existe partícula o tipo de radiación cuya curva de deposición de dosis en profundidad esté completamente localizada, por lo que inevitablemente, a la hora de aplicar radiación a un tumor con el fin de provocar la muerte de sus células, el tejido sano también se ve expuesto a la radiación.

El objetivo de la radioterapia es, por tanto, conseguir una distribución de dosis absorbida uniforme en el volumen blanco, donde se encuentran las células cancerosas, y una dosis absorbida tan baja como sea posible en los tejidos sanos alrededor de ese volumen.

En la mayoría de los casos, la dosis total de radiación necesaria para combatir el tumor no puede suministrarse por completo en una sola sesión. Esto se debe a que una gran dosis de radiación administrada en una misma sesión puede causar más daños a los tejidos sanos adyacentes, lo que puede causar más efectos secundarios que si se administra esa misma dosis distribuida en muchas sesiones de tratamiento durante varios días o semanas [19].

Según el tipo de cáncer, el tamaño y la ubicación del tumor en el cuerpo del paciente, así como su cercanía a órganos sanos, existen tres tipos principales de radioterapia: radioterapia de haz externo, braquiterapia y radioterapia intraoperatoria. La modalidad más extendida es la radioterapia de haz externo o radioterapia externa en la que se emplean máquinas denominadas aceleradores lineales, que aceleran partículas cargadas hasta altas velocidades. Estas partículas al frenarse en un medio pueden generar haces intensos de fotones que pueden llegar a ser muy energéticos. Los haces de fotones o de las partículas aceleradas son dirigidos hacia el interior del cuerpo, atravesando el tejido hasta alcanzar el volumen blanco.

Dentro de la radioterapia externa, existen diferentes técnicas de tratamiento [20] (utilizaremos el nombre en inglés por comodidad):

- **Intensity Modulated Radiation Therapy (IMRT):** se utilizan haces de radiación con intensidades variables para adecuar la dosis de radiación a la forma y situación del tumor. En esta técnica la máquina se va moviendo con determinados ángulos y emitiendo la radiación de forma estática [21].
- **Volumetric Modulated Arc Therapy (VMAT):** el haz de radiación gira alrededor del paciente, modulando la intensidad de la radiación y la velocidad del giro. El haz de radiación se adapta a la forma del tumor y se administra desde todos los ángulos mientras el acelerador lineal gira alrededor del paciente por lo que evita una alta dosis de irradiación en el tejido sano [22].
- **Stereotactic radiosurgery (SRS):** se diferencia de las dos anteriores en que entrega mayores dosis de radiación en un menor número de sesiones de tratamiento (entre 1 y 5 sesiones). Gracias a las mejoras tecnológicas, tanto de los aceleradores como de los sistemas de imagen, en esta técnica, tanto el volumen contorneado como la dosis suministrada, quedan muy conformados en el tumor, por lo que se conserva el tejido sano. Está enfocada a patologías intracraneales [23].
- **Stereotactic Body Radiation Therapy (SBRT):** se denomina SBRT a la técnica SRS utilizada en tumores ubicados en partes del cuerpo fuera del cráneo.

### 1.3.2. Linear Accelerators (LINACs)

Los LINACs son máquinas que aceleran electrones a energías cinéticas en el rango de 4 a 25 MeV usando campos de radiofrecuencia<sup>1</sup> (RF). A diferencia de las máquinas más tradicionales, los LINACs no disponen de fuente radiactiva (como el Co-60). Un acelerador lineal típico moderno, de alta energía, provee fotones de dos energías (alrededor de 6 y 18 MV) y electrones de distintas energías (por ejemplo 6, 9, 12, 16 y 22 MeV) [24].

En su forma más básica una máquina de este tipo es un dispositivo que acelera partículas cargadas (generalmente electrones) y luego las hace impactar contra un blanco, donde se genera el haz de fotones que se utilizarán para irradiar los tumores. A continuación, se presenta un esquema de un acelerador lineal y se expone una descripción de sus partes y su función.

<sup>1</sup>En el rango de las microondas ( $10^3$  MHz -  $10^4$  MHz)

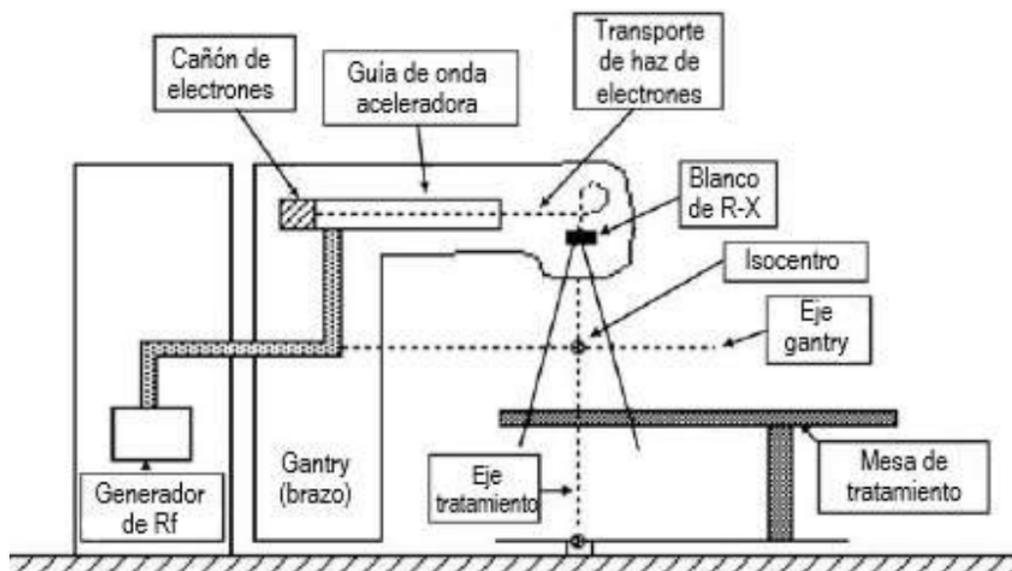


Figura 1.4: Esquema de un acelerador lineal de uso médico. Fuente: formaciontss.es [25].

- (I) **Sección generadora.** Está constituida por el cañón de electrones y el generador de RF.

El cañón de electrones contiene un filamento metálico, típicamente tungsteno, que emite electrones termoiónicamente al ser calentado. Estos electrones ingresan en un tubo provisto de varias cavidades (guía aceleradora) donde son acelerados mediante campos electromagnéticos. El número de electrones ingresados en la sección aceleradora es controlado por la temperatura del filamento.

Estos campos de RF se producen en el generador de RF mediante el proceso de desaceleración de electrones en potenciales de retardo en dispositivos especiales llamados magnetrones o klystrons, según el fabricante. La potencia y frecuencia de estas ondas determina la energía final de los rayos X.

- (II) **Sección aceleradora.** Consiste en una serie de discos con agujeros circulares en el centro, ubicados equidistantes y que, en conjunto, forman un tubo cilíndrico uniforme. Estos discos dividen la guía de ondas en una serie de cavidades en la cual se ha realizado el vacío para evitar las desviaciones de los electrones.

Existen dos tipos de guías de ondas para la aceleración de electrones: estructura de onda estacionaria y estructuras de onda viajera o progresiva. En las estructuras de onda estacionaria, las microondas rebotan al final del cañón por lo que el campo interno es más intenso y los electrones adquieren mayor energía en un menor espacio. En las estructuras de onda progresiva esto no sucede, por lo que necesita una mayor longitud de cañón. Las estructuras de onda progresiva son más sencillas de construir.

La longitud de la guía de ondas depende de la energía cinética final del electrón requerida, y oscila entre 30 cm para una energía de 4 MeV y 150 cm para una de 25 MeV.

- (III) **Sistema de transporte del haz.** La guía de ondas aceleradora está usualmente montada paralela al eje de rotación del gantry<sup>2</sup> por lo que el haz de electrones debe ser re-direccionado para impactar en el blanco de rayos X. Para lograr el re-direccionamiento, existe, en LINACs que trabajan a energías por encima de 6 MeV, un sistema de imanes ubicado muy cerca del extremo final de la guía.

Este conjunto de imanes suele denominarse sistema magnético de deflexión del haz y existen tres configuraciones disponibles para el ángulo de re-dirección:  $90^\circ$ ,  $112,5^\circ$  y  $270^\circ$ . En la figura 1.4 se muestra un sistema de deflexión a  $270^\circ$ .

- (IV) **Cabezal.** En la Figura 1.5 se muestran los diferentes componentes del cabezal.

Una vez que el haz de electrones atraviesa el vacío e impacta en el blanco, compuesto por tungsteno y cobre, se produce una radiación de frenado y en consecuencia rayos X. Si se quiere tratar al paciente con electrones directamente, puede quitarse el blanco. Tras el blanco, si es que lo hay, se sitúa un bloque de tungsteno en el que se ha mecanizado una abertura cónica que actúa como colimador primario. A continuación, ya que la intensidad del haz no es homogénea, necesita ser homogeneizada para su uso clínico, se emplea un filtro aplanador.

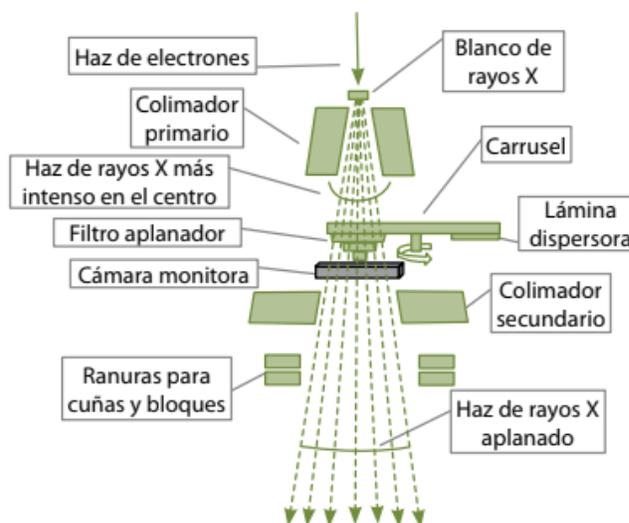


Figura 1.5: Esquema del cabezal de un acelerador para producir RX. Fuente: Fundamentos Física Médica Volumen 3 [26].

Seguido del filtro, el haz pasa por una cámara de ionización de transmisión denominada cámara monitora. Esta controla la dosis y la tasa de dosis administrada durante los tratamientos. El LINAC se calibra para que 1 cGy corresponda con 1 unidad monitor (UM), magnitud con la que se trabaja.

Finalmente el colimador secundario trata de colimar el haz de radiación mediante unas mordazas de plomo. Además, los LINACs modernos cuentan con el Colimador Multi-lamina (MLC), que es lo último que ve el haz de rayos X antes de entrar en contacto con el paciente. Está constituido por una serie de láminas plomadas que se abren y cierran de forma independiente para permitir o no el paso de radiación. Gracias a estas láminas, el haz puede adquirir la forma de una sección del tumor, minimizando enormemente la energía depositada en el tejido sano circundante. En la figura 1.6 se puede comprobar la eficacia de conformación de las multiláminas para un cáncer de cabeza y cuello.

En un LINAC se tienen tres ejes de rotación fundamentales e independientes para el cabezal

<sup>2</sup>Brazo móvil del acelerador lineal

o gantry, el colimador y la mesa de tratamiento, en la que el paciente se posiciona para ser tratado. Se denomina isocentro al punto en el espacio donde el gantry, el colimador y la mesa rotan en común (figura 1.4).

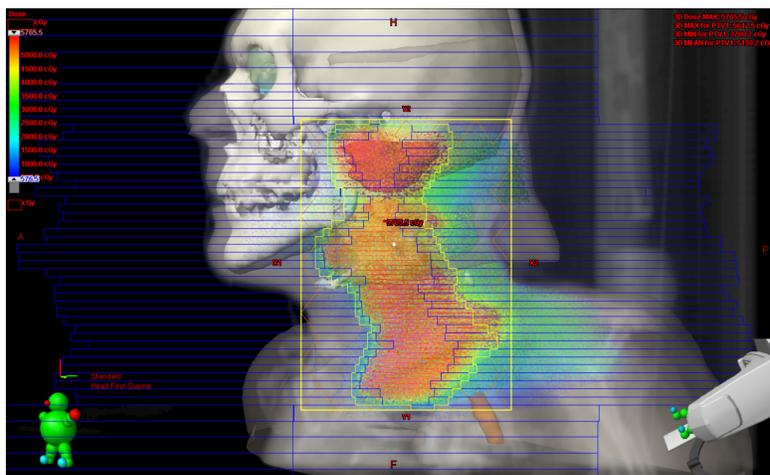


Figura 1.6: Ejemplo del uso de las multiláminas (azul) para un tumor localizado en la garganta. Las zonas más rojas indican mayores dosis recibidas y el muñeco verde la posición del paciente.

### 1.3.3. Etapas del proceso radioterápico

Desde que un paciente acude a un Servicio de Radioterapia se sigue con él un proceso complejo, atendido por diferentes profesionales, que comprende varias etapas que son descritas a continuación [27]:

- (a) **Decisión terapéutica y prescripción.** Se determina el objetivo y la modalidad terapéutica (radioterapia externa, braquiterapia, radioterapia intraoperatoria). Además se incluye la dosis total a administrar, el fraccionamiento (dosis por sesión, periodicidad y duración del tratamiento) y tipo de radiación.
- (b) **Adquisición de imágenes TC e inmovilización del paciente.** Se toman imágenes TC (tomografía computarizada) con el paciente en la camilla o mesa, tal y como se le va a posicionar los días de tratamiento. Los técnicos de radioterapia deben colocar al paciente adecuadamente, procurando que se mueva lo menos posible, y en una posición reproducible. A continuación, se establecen unas marcas de referencia colocando unos marcadores o puntos tatuados (PTs) sobre el paciente, fabricados de materiales radioopacos para que tengan un alto contraste y sean fácilmente diferenciables en la imagen.

Los marcadores radioopacos son de vital importancia ya que establecen un sistema de referencia sobre el paciente virtual (imágenes TC). Durante la planificación de un tratamiento sobre el paciente virtual, se establece que el origen de coordenadas (0, 0, 0) queda determinado por las posiciones de los tres marcadores. Por tanto, las coordenadas del isocentro calculadas del plan de tratamiento (punto próximo al tumor) dependen directamente de la posición de dichos PTs en el TC.

El LINAC cuenta con un sistema de láseres que forman crucetas sobre el paciente y que indican el isocentro del acelerador. En la sesión de tratamiento, cuando el paciente se coloca en la camilla, se hacen coincidir los marcadores con las crucetas de los láseres del acelerador. Así, se tiene que el isocentro del plan tiene coordenadas  $(0, 0, 0)$ . En caso de que el isocentro planificado tenga coordenadas no nulas, por ejemplo  $(0.2, 3.0, 7.2)$ , se desplazaría la camilla 0.2 cm en lateral, 3 centímetros en vertical y 7.2 cm en longitudinal. Así, el isocentro de la máquina queda próximo al tumor.

- (c) **Importación de las imágenes TC, del paciente en el sistema de planificación.** Estas imágenes son la base de la planificación del tratamiento radioterápico ya que son nuestro paciente virtual.
- (d) **Definición de la anatomía.** A partir de las imágenes, un oncólogo radioterapeuta debe definir y visualizar contornos y superficies de estructuras, como pueden ser los órganos sanos, el tejido tumoral o la propia camilla, así como generar el volumen 3D del paciente. Para definir el volumen del tumor se utilizan dos estructuras: CTV (Clinical Target Volume), que es la región que contiene el tumor, y PTV (Planning Target Volume), que es una expansión adicional del CTV que tiene en cuenta las incertidumbres en la administración de la radioterapia como pueden ser los márgenes de movimiento del paciente.
- (e) **Prescripción de la dosis a administrar al tumor y de los límites de dosis para los órganos adyacentes.** El oncólogo radioterápico asigna la dosis que debe recibir el tumor así como la dosis límite para cada órgano sano involucrado en el medio circundante. Existen numerosas guías para prescribir la dosis según la patología a tratar.
- (f) **Planificación del tratamiento en el sistema de planificación de tratamientos (TPS):** Cada TPS utiliza un algoritmo de cálculo, estos, mediante un conjunto de reglas y procedimientos matemáticos, son capaces de reunir, organizar y sintetizar grandes volúmenes de datos. Toman en cuenta muchos factores, como la forma y el tamaño del tumor, la anatomía del paciente, la respuesta tisular a la radiación y otros parámetros clínicos y físicos. Habiendo determinado el algoritmo a utilizar, éste permite determinar la dosis que se le va a suministrar al paciente.
- (g) **Evaluación del plan.** Se generan vistas de dosis absorbida en dos dimensiones (2D) y tres dimensiones (3D) y, haciendo uso de herramientas como pueden ser los histogramas Dosis-Volumen (HDV), se determina la eficacia del tratamiento. Un histograma dosis-volumen es un gráfico de distribución de dosis acumulada con respecto al volumen, que resume gráficamente la distribución de la radiación simulada dentro de un volumen de interés de un paciente [28]. Haciendo uso de este, se comprueba si se sobrepasan o no los límites de dosis establecidos.

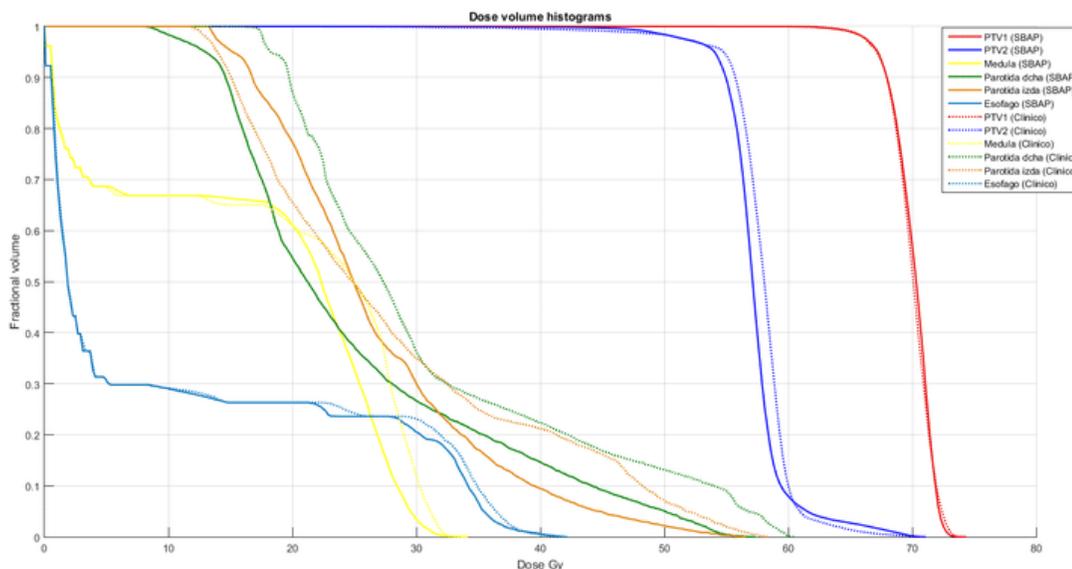


Figura 1.7: Ejemplo de histograma Dosis-Volumen. Fuente: ResearchGate [29].

En el histograma dosis-volumen previo aparecen estructuras como las glándulas parótidas, el esófago o la médula espinal. La estructura PTV (del inglés “Planning Target Volume”) representa el volumen que se pretende tratar con radiación y está diseñada para abarcar el volumen del tumor, así como una posible incertidumbre.

La cuantificación de la radiación absorbida por los órganos de riesgo se puede expresar mediante el porcentaje de volumen que recibe una dosis determinada, por ejemplo  $V_{25Gy} < 10\%$ : la dosis que recibe un 10% del volumen debe ser inferior a 25 Gy. Cada órgano tiene sus propios límites, los cuales se encuentran tabulados [30]. Por ejemplo, en el caso del esófago  $V_{20Gy} < 30-35\%$  y la dosis máxima debe ser inferior a 45-60 Gy. En la Figura 1.7 ambas se cumplen, por lo que la dosis que recibe el esófago en ese tratamiento está permitida.

Como es de esperar, realizar este procedimiento cuando son muchos órganos los implicados, como puede suceder en tumores ubicados en la cabeza, resulta un trabajo muy tedioso.

- (h) **Revisión del plan.** Antes de implementar el plan, deben revisarse aspectos de la planificación para evitar errores en la administración de la dosis al paciente. De esto se hablará más en los próximos apartados, pues es el punto en el que se ha centrado el trabajo.
- (i) **Administración del tratamiento.** El plan es enviado al LINAC para que comiencen las sesiones de tratamiento. Los técnicos de radioterapia posicionan al paciente en la mesa de tratamiento, haciendo coincidir sus PTs con el sistema de láseres del LINAC.

# Capítulo 2

## Hipótesis y objetivos

Dado el efecto de las radiaciones ionizantes, es de vital importancia tener la seguridad de administrar correctamente la dosis de tratamiento a los pacientes. El Servicio de Oncología Radioterápica (SOR) del Hospital Universitario Marqués de Valdecilla (HUMV) recoge todos los aspectos que deben verificarse en una “check-list” o lista de comprobación para finalmente administrar el tratamiento en el LINAC. Es responsabilidad del radiofísico hospitalario la revisión del cumplimiento de todos los requisitos por lo que está sujeta a posibles errores humanos.

Por un lado, en la etapa de verificación de los tratamientos se comprueban numerosos aspectos cruciales del plan diseñado, como el nombre del paciente y su curso de tratamiento, la dosis y el número de sesiones de prescripción, o la correcta definición del isocentro de radiación. Se deben revisar numerosos datos y parámetros de la planificación que suponen una labor repetitiva y un gasto de tiempo significativo. Asimismo, el análisis de los histogramas dosis volumen (HDV) para la comprobación de los límites de dosis de los órganos de riesgo y los volúmenes blanco es una estrategia sencilla y eficaz, pero resulta en un trabajo laborioso cuando el número de volúmenes es elevado, como suele suceder en los pacientes con cáncer de cabeza y cuello o en tratamientos de SBRT.

El sistema de planificación Eclipse ofrece al usuario la posibilidad de crear programas informáticos (*scripts*) con la ayuda de una interfaz de programación o API (Application Programming Interface) que explora la información dosimétrica del plan. El lenguaje de programación que emplea es C#, que es orientado a objetos.

Con el fin de facilitar las labores en tratamientos complejos, disminuir el tiempo empleado por paciente y reducir, en lo posible, el error humano, se va a diseñar un *script* en C# para que verifique de forma automática los pasos de la check-list, entre ellos el análisis dosimétrico. Como es evidente, el objetivo del código no es realizar completamente la verificación del plan, sino más bien servir como punto de apoyo al profesional en cuestión.

# Capítulo 3

## Material y método experimental

### 3.1. Red ARIA

El sistema de información oncológica ARIA, desarrollado por la compañía *Varian Medical Systems*, se utiliza para la gestión de los tratamientos radioterápicos del SOR del HUMV. La red ARIA se considera como una gran base de datos que recoge toda la información del paciente radioterápico, como datos del paciente, imágenes clínicas, prescripciones de dosis, planificaciones y programaciones de tratamientos y resultados de los controles de calidad de dichos tratamientos, en una única plataforma intuitiva y de fácil uso para los profesionales. Esta red documenta las actividades del proceso radioterápico y ayuda a los profesionales a optimizar la atención al paciente, pudiendo acceder a sus datos de forma sencilla (Página oficial de ARIA: [31]).

Se ha utilizado la red ARIA para seleccionar pacientes anónimos, tratados con anterioridad de cáncer de cabeza y cuello, de pulmón y de huesos y acceder a sus prescripciones de dosis y sus planes de tratamientos.

### 3.2. Planificador Eclipse

Se emplea el sistema de planificación de tratamientos (TPS) Eclipse v.16.1 de *Varian Medical Systems*, el cual está integrado dentro de la red ARIA, para la simulación de los haces de radiación y el cálculo de la distribución de la dosis en el paciente [32]. Los TPS poseen la capacidad de calcular la dosis mediante algoritmos matemáticos que permiten predecir las distribuciones de dosis en un paciente a partir de mediciones experimentales. En particular, el planificador Eclipse dispone de diferentes licencias de planificación (3D, IMRT, VMAT) y algoritmos de cálculo de dosis (AAA, Acuros) para adecuar el plan de tratamiento a cada paciente (Página oficial de Eclipse: [33]).

En la Figura 3.1 se muestra la vista general de una planificación de un tumor de cabeza y cuello en la interfaz de Eclipse. Se observa la distribución de dosis calculada en las vistas axial (superior

izquierda), coronal (inferior izquierda) y sagital (inferior derecha) sobre las imágenes TC de un paciente. También se muestran los contornos del tumor (CTV, PTV) y de los órganos de riesgo (cavidad oral, mandíbula, esófago, cerebro, etc.), los cuales deben cumplir las limitaciones de dosis propuestas en la prescripción médica.

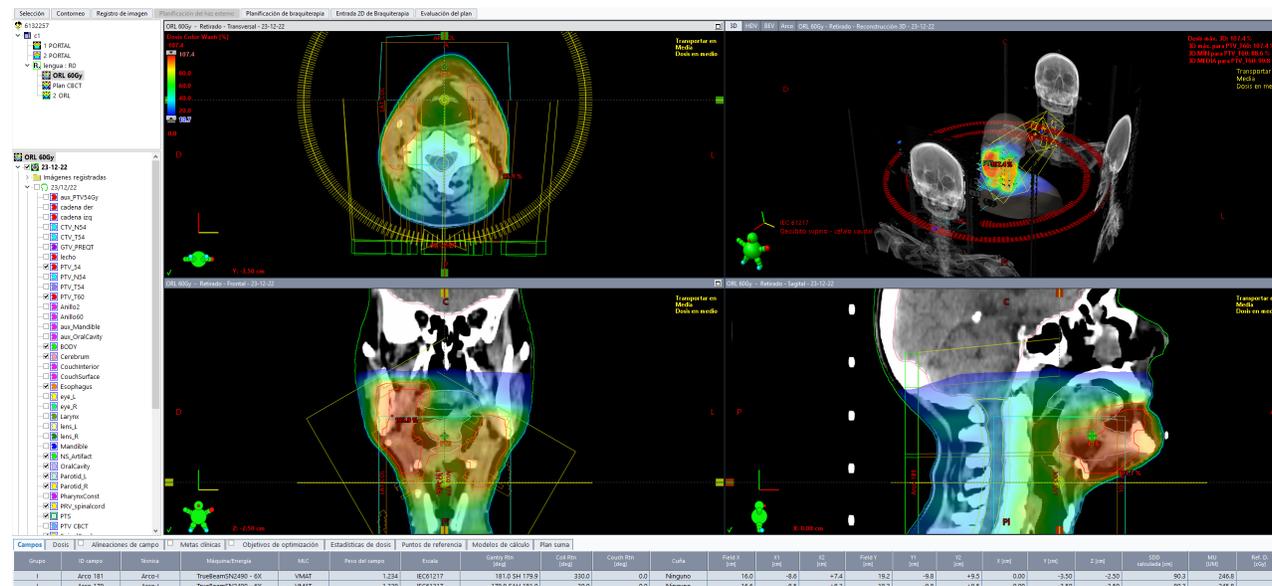


Figura 3.1: Interfaz del planificador de Eclipse para un tumor ubicado en la cavidad oral (ver más detalles en el texto).

En la parte inferior se muestran diferentes secciones, como “Campos” o “Dosis”, en las que se almacena la información del diseño del plan y el esquema de dosis. En el contexto de la radioterapia, un campo se refiere a la configuración específica de la radiación utilizada para tratar un área particular del cuerpo del paciente. En la Figura 3.1 tenemos dos campos denominados “Arco 181” y “Arco 179” que se encargan de irradiar de  $0^{\circ}$  a  $179^{\circ}$  en sentidos opuestos. Por tanto, en el apartado “Campos” se recogen los datos como el número y la nomenclatura de estos, el acelerador y la energía de la radiación utilizada, la técnica elegida, las angulaciones del gantry, del colimador y de la camilla, las coordenadas del isocentro, las unidades monitor por cada uno, etc. En el apartado “Dosis” se muestra la información de la prescripción de dosis como la dosis total, el número de fracciones, la dosis por fracción, etc.

El *script* que se diseña para la automatización de la verificación de los tratamientos navega a través de estas pestañas con el fin de acceder y revisar datos y parámetros determinantes que garanticen la idoneidad del plan de tratamiento para su posterior administración.

### 3.3. Eclipse Scripting API

Eclipse Scripting API (ESAPI) es una interfaz de programación de aplicaciones (API) integrada en el TPS Eclipse. Esta API ofrece al usuario la posibilidad de crear *scripts* y proporciona herramientas y funcionalidades que los desarrolladores pueden utilizar para crear sus propios

complementos, plugins o herramientas, mediante un lenguaje de programación compatible, como C#.

ESAPI ofrece un conjunto de clases, objetos, funciones, etc., de Microsoft.NET que permite el acceso a la lectura de datos y parámetros de una planificación de un tratamiento de radioterapia externa realizada con el TPS Eclipse y que devuelve información sobre el diseño del plan (número de campos, técnica utilizada, coordenadas del gantry, de colimador, de la camilla y del isocentro, etc.), el esquema de dosis prescrito (dosis total al tumor, número de sesiones, dosis por fracción, etc.) o el DVH, entre otros.

ESAPI se puede utilizar de dos maneras diferentes desde el punto de vista de interacción con el usuario [34]: creando *Standalone Executable Scripts*, que se ejecutan como aplicaciones ejecutables de Windows, o bien, creando *Plug-in Scripts*, que se ejecutan desde la sesión de Eclipse dando acceso únicamente al paciente tratándose en ese momento. Este último modo es el elegido en este trabajo.

El *script* se ha diseñado utilizando las librerías<sup>1</sup>, las clases<sup>2</sup>, las subclases y los atributos recogidos en la guía de ayuda *Eclipse Scripting API Help* (Figura 3.2), la cual consta de más de 150 clases y subclases con decenas de atributos.

The screenshot shows the Eclipse Scripting API Help interface. At the top, it says "The PlanSetup type exposes the following members." Below this, there are two sections: "PlanSetup" and "Beams". Each section contains a table with "Name" and "Description" columns.

PlanSetup	
Name	Description
ApplicationScriptLogs	The log entries of the script executions that have modified the pl...
ApprovalHistory	Returns the approval history of the plan setup.
ApprovalStatus	The approval status.
BaseDosePlanningItem	BaseDose for the optimization, can be either plan or plan sum.
Beams	A collection of all the beams in the plan (including setup beams). the plan is a brachytherapy plan.
Comment	[Availability of this property depends on your Eclipse Scripting AF

Beams	
Name	Description
Applicator	An applicator is a specific add-on used in the beam.
ArcLength	The arc length.
AverageSSD	The average Source-to-Skin Distance (SSD) of an arc beam.
Id	The identifier of the Beam.
IsGantryExtended	Checks if the gantry rotation is extended. For arc beams, checks if the gantry rotation
Boluses	A collection of beam boluses.

Figura 3.2: Interfaz de *Eclipse Scripting API Help* donde se recoge un listado de atributos de la subclase *Beams* que a su vez está contenida dentro de la clase *PlanSetup*. La subclase *Beams* contiene muchos de los valores que se pueden encontrar en el apartado “Campos” (Figura 3.1), que recogen las características principales de cada haz de radiación. Por ejemplo, si se quiere acceder al nombre de un campo de un plan de tratamiento, dentro de la subclase *Beams* se tiene el atributo denominado *Id* que, según su descripción, se corresponde con el identificador de cada campo. Así pues, escribiendo en código “PlanSetup.Beams.Id” se navega a través de las secciones del plan, se accede a la pestaña “campos” y se lee y extrae la información de la nomenclatura del campo.

### 3.4. *Check-list* de verificación de los tratamientos

La planificación de un tratamiento no es válida hasta que se verifiquen determinados requisitos relacionados con el diseño y la dosimetría del plan. En el SOR del HUMV, estos parámetros se recogen en una lista de comprobación o *check-list* para facilitar su revisión. A continuación, se explican cada una de las tareas del *check-list*:

<sup>1</sup>Conjunto de código preescrito que contiene funciones, clases y rutinas utilizables para desarrollar software.

<sup>2</sup>Agrupación de datos (llamados atributos) y funciones (llamadas métodos) relacionados.

1. **Consultar notas.** Las notas son mensajes escritos por los profesionales para dejar constancia de información relevante para el tratamiento del paciente. Estas notas deben ser revisadas con el fin de no pasar por alto dicha información (por ejemplo, se puede indicar si tratar al paciente es una urgencia o no, si han sucedido cambios en alguna etapa, etc.)
2. **Abrir láminas.** Este paso únicamente es aplicable para tratamientos de mama. Para contemplar la posible inflamación de las mamas tras varias sesiones de irradiación, las láminas del MLC se abren hacia el aire manteniendo la forma ovalada de la mama (Figura 3.3).

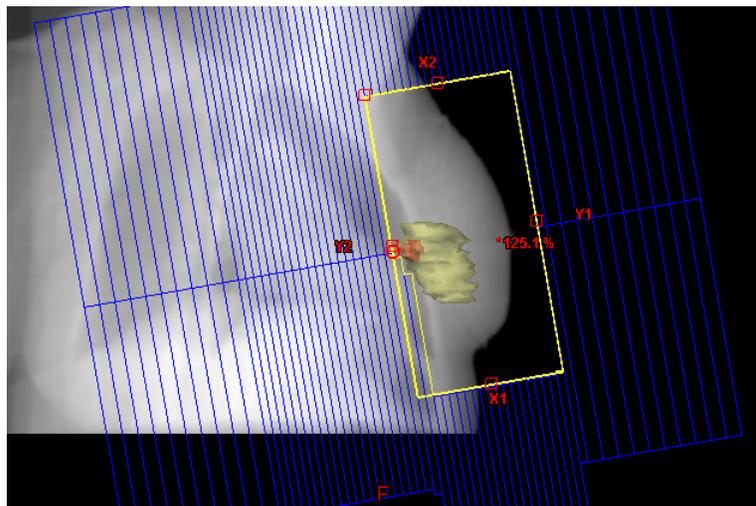


Figura 3.3: Punto de vista de un haz de un plan de tratamiento de cáncer de mama. El MLC (azul) se abren siguiendo la forma de la mama en la zona del aire para contemplar la inflamación de la mama tras varias irradiaciones.

3. **Correspondencia del nombre del paciente, prescripción y etapa de tratamiento.** Dado el efecto de las radiaciones ionizantes, es de suma importancia que la identidad del paciente corresponda con las imágenes TC sobre las que se planifica, con las prescripción de dosis y con el curso de tratamiento en el que está inmerso. Un mismo paciente puede tener cursos de tratamiento anteriores ya administrados y es de vital importancia evitar confusiones con sus imágenes TC y prescripciones de dosis más antiguas.
4. **Verificación del origen PTs.** Las proyecciones de los tres marcadores radioopacos (PTs), colocados durante la simulación TC, indican las coordenadas del origen del plan de tratamiento por lo que deben ser visibles en las imágenes TC de planificación e indicar el origen (0,0,0). Este paso es fundamental ya que el paciente se coloca en la camilla de tratamiento mediante los PTs y los láseres del acelerador (ver paso b de “Etapas del proceso radioterápico”).
5. **Verificación de la dosis de prescripción y del número de sesiones correcto.** Se comprueba que el número de sesiones y la dosis prescrita coincide con la el número de sesiones y la dosis introducida en el planificador para el cálculo del plan tratamiento

6. **Correcta posición del paciente y mesa de tratamiento.** El cálculo de la deposición de dosis debe calcularse teniendo en cuenta la camilla sobre la que se posiciona al paciente, ya que es un material absorbente de dosis a tener en cuenta. Por tanto, es crucial haber introducido en el plan de tratamiento las estructuras *CouchInterior* y *CouchSurface*, que corresponden con la mesa de tratamiento de un paciente.
7. **Técnica de tratamiento adecuada según la prescripción.** Se comprueba que la técnica empleada es VMAT, donde la nomenclatura de los campos es “Campo 1”, “Campo 2”, etc. o IMRT, donde el nombre se corresponde con el ángulo para el cual se irradia sin el símbolo de grados (90, 270, etc.).
8. **Posición correcta del isocentro.** Se comprueba que el isocentro se coloca centrado aproximadamente en el tumor y que sus coordenadas están redondeadas, pues facilita el trabajo a los técnicos de radioterapia cuando posicionan al paciente en la mesa de tratamiento.
9. **Todos los campos asociados al mismo acelerador y con el mismo isocentro.** Los campos de un plan de tratamiento deben tener el mismo isocentro (salvo excepciones, como irradiaciones creaneo-espinales) para simplificar el trabajo y acortar el tiempo total de una sesión de tratamiento. Todos los haces de radiación son suministrados en un mismo acelerador por lo que todos los campos de un plan de tratamiento deben estar asociados a un mismo acelerador.
10. **Matriz de dosis correcta.** Según qué patología se trate, la malla de cálculo<sup>3</sup> es de mayor o menor resolución. En general, la resolución del cálculo de la dosis en los planes de tratamiento es de 2,5 mm, sin embargo, para el caso de SBRTs debe seleccionarse 1,25 mm.
11. **Verificación del algoritmo de cálculo.** Por defecto, se emplea el algoritmo de cálculo “Acuros XB” pues permite un cálculo de dosis ante heterogeneidades del medio más adecuado. Sin embargo, en ocasiones se utiliza “Anisotropic Analytical Algorithm” (AAA). Se quiere comprobar si el plan de tratamiento utiliza uno u otro.
12. **Verificación de que no hay colisiones del acelerador con la mesa, los soportes y/o el paciente.** Cuando la camilla de tratamiento está rotada puede dar lugar a colisiones con el gantry. Esto empeora cuando, por ejemplo, el paciente se posiciona con los brazos flexionados tras la nuca, cuyos codos también pueden colisionar. En general, y previamente a la primera sesión de tratamiento de un paciente, suelen simularse los movimientos del gantry, teniendo en cuenta tanto la rotación de la camilla como la posición del paciente (y sus inmovilizadores). Sin embargo, automatizar esta tarea nos recuerda la probabilidad no nula de colisión. Se debe leer el ángulo de rotación de la camilla (*Couch Rotation*) y siempre que sea distinto de cero, avisar con un mensaje de alerta y valorar las posibilidades de colisión.
13. **Verificación de que se cumplen los objetivos y restricciones.** Este paso es uno de los más relevantes dentro del check-list y consiste en el cumplimiento de la prescripción de la dosis propuesta por el oncólogo radioterápico. Tanto el tumor como los órganos de riesgo

---

<sup>3</sup>Técnica para calcular la distribución de dosis de radiación en el tejido.

deben cumplir ciertos límites de dosis en función de guías internacionales médicas para cada tipo de patología. Se debe explorar el histograma dosis-volumen para cada órgano y para el tumor y comprobar que se cumplen dichas restricciones.

14. **La distribución de dosis correcta por plano, por plan y por el tratamiento completo.** Se debe comprobar que la distribución de dosis por plano es adecuada, es decir, que la dosis de prescripción quede confinada entorno al tumor y que los órganos de riesgo no reciban dosis superiores a las indicadas. No deben aparecer máximos de dosis de prescripción fuera del tumor en ninguno de los planos, es decir, en ninguno de los cortes del TC. En ocasiones, un mismo tratamiento consta de dos planes: uno que busca irradiar al tumor y otro que suministra un extra de dosis a cierta zona del tumor donde se sospecha que haya más restos tumorales (boost). La distribución de dosis por plano de cada plan, así como la del plan total, que es la suma de ambos, debe presentar una distribución de dosis adecuada.
15. **Verificación de que no se exceden los límites de UM de la máquina.** Se debe comprobar que en ningún campo se excede el límite de 1500 UM (ver sección “Cabezal” de “Linear Accelerators (LINACs)”).
16. **Colocación correcta de Bolus: posición, densidad, espesor y asignación de haces.** En la Figura 1.2 se representa la profundidad a la que la deposición de la dosis es máxima para cada tipo de radiación. Los fotones de 6 y 18 MeV, presentan máximos de deposición de dosis a 1.5 y 3 cm aproximadamente, lo cual puede no ser adecuado para el tratamiento de tumores superficiales (mama, piel, etc.). Para mejorar esto se utiliza el bolus, un material con propiedades equivalentes al tejido humano cuando es irradiado [35], que se coloca en contacto con la parte superficial del tumor. El bolus permite que el máximo de deposición de dosis indicada por la curva de deposición coincida con la posición del tumor. En este paso, se debe comprobar que la posición, la densidad y el espesor del bolus es adecuado y que está asignado a cada uno de los campos de radiación para garantizar que el cálculo de deposición de dosis es correcto.



Figura 3.4: Uso de bolus para tratar un tumor en el cuello. Fuente: ScienceDirect [36].

### 3.5. Diseño del *script*

El *script* se implementa en lenguaje de programación C# con la ayuda de Visual Studio, un entorno de desarrollo integrado (IDE) para Windows recomendado por *Varian* y compatible con múltiples lenguajes de programación que facilita la edición, depuración y compilación de los códigos de programación. Sin embargo, se pueden escribir en cualquier editor de texto, ya que al ejecutarlos desde la propia interfaz de Eclipse no necesitan ser compilados.

El *script* se organiza en diferentes secciones:

1. En el primer fragmento del código se muestra el encabezado, en el cual se cargan un conjunto de librerías que son usadas por las funciones que aparecen más adelante. De todas las librerías, cabe destacar las correspondientes a las líneas 9 y 10 (MS.TPS.Common.Model.API y MS.TPS.Common.Model.Types), necesarias para usar ESAPI.
2. El cuerpo principal del *script* corresponde a las funciones creadas para automatizar los pasos del check-list de verificación de tratamientos. En particular, el código para la definición de las funciones del *script* es:

```
static string NombreFunción(ref Argumento Abreviatura Argumento)
```

Por ejemplo, dentro de la clase *PlanSetup*, cuya abreviatura es “ps”, es posible acceder a las notas o comentarios de un plan mediante el atributo *Comment*. Si la función se nombra como *CheckNotas*, entonces su código corresponde a:

```
static string CheckNotas (ref PlanSetup ps)
```

Dentro de esta función, basta con escribir “return ps.Comment”, para que la función devuelva la información contenida en el atributo *Comment*.

3. A continuación, el *script* comprueba que el contexto sea correcto, es decir, que exista un paciente y un plan seleccionado para navegar a través de su información. Como ya se ha mencionado, se trabaja en el modo *plug-in script*, integrada en la propia interfaz de Eclipse. Es necesario trabajar con un contexto, es decir, toda la información que se maneja a través de ESAPI hará referencia al contexto de Eclipse que esté seleccionado en el momento de ejecutar el *script*. Por tanto, si se tiene seleccionado un plan de Eclipse, *context.PlanSetup* se refiere a dicho plan.

Además se definen *context.StructureSet*, *context.PlanSetup*, *ss.Image* y *ps.RTPrescription* como *ss*, *ps*, *image* y *rtPrescription*, respectivamente. Con esto, si en el código se quiere utilizar alguna de estas clases o subclases basta con poner su abreviatura y no el recorrido completo para acceder a ellas.

4. El resto del código se corresponde con la creación de dos ventanas emergentes. La primera ventana emergente es la unión de los resultados de ejecutar todas las funciones creadas menos una, que es mostrada en la segunda ventana emergente por devolver mayor cantidad de información.

### 3.6. Pacientes para comprobar el código

El script diseñado se ejecuta para diferentes pacientes con diversas patologías. En particular, en este trabajo, se muestran los resultados para tres pacientes que, por respetar la confidencialidad, denominaremos A, B y C, que presentan patologías de cabeza y cuello, pulmón y vértebras, respectivamente. Los tres pacientes seleccionados han sido tratados con fotones.

#### Paciente A.

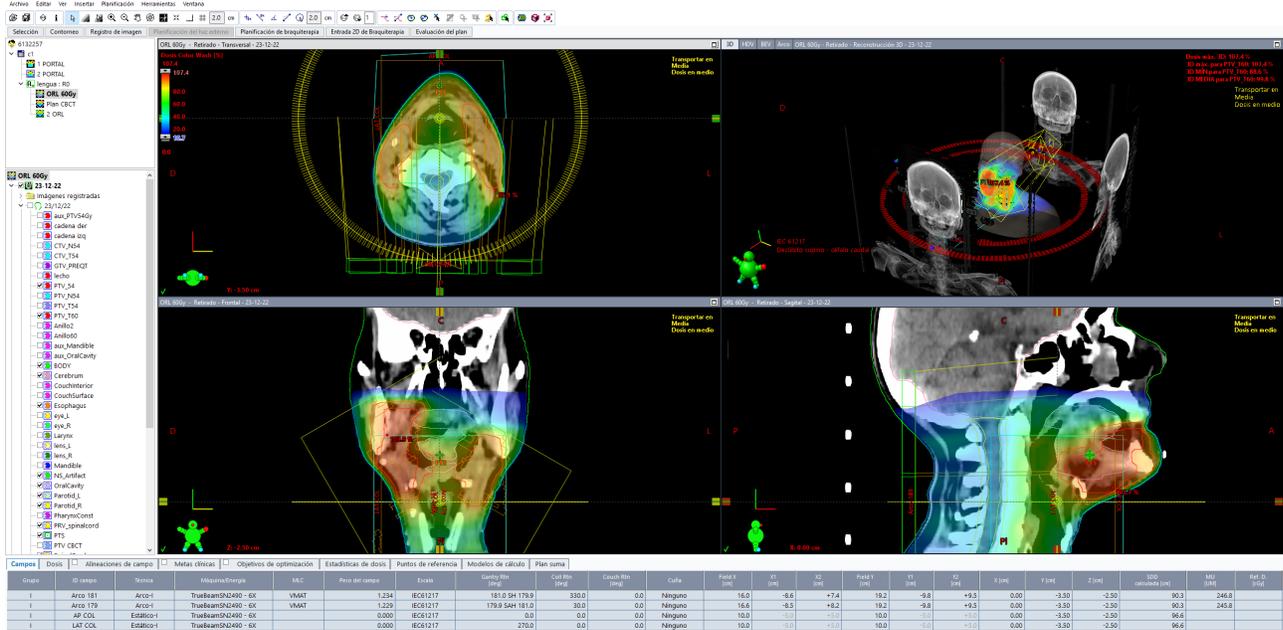


Figura 3.5: Captura de pantalla del planificador para el Paciente A.

El plan de tratamiento del paciente A (Figura 3.5) se caracteriza por:

- Únicamente consta de un curso de tratamiento.
- Tumor de cabeza y cuello.
- Estructuras contorneadas (cuerpo, cerebro, esófago, laringe, ojos, cristalinos, glándulas parótidas, faringe, etc.).
- Dispone de las estructuras *CouchInterior* y *CouchSurface*, que conforman la camilla de tratamiento del paciente.

Campos	Dosis	Alineaciones de campo	Metas clínicas	Objetivos de optimización	Estadísticas de dosis
ID del plan	Dosis por fracción [cGy]	Número de fracciones	Dosis total [cGy]		
ORL 60Gy	200.0	30	6000.0		

Figura 3.6: Dosis total, dosis por sesión y número de sesiones para el Paciente A.

- Prescripción de dosis (Figura 3.6): el tumor debe recibir 6000 cGy en 30 sesiones de 200 cGy.
- Acelerador: TrueBeamSN2490.
- Técnica de tratamiento: VMAT (dos arcos completos, de  $0^{\circ}$  a  $179^{\circ}$ , en sentidos opuestos).
- Mesa de tratamiento sin rotación.
- Unidades de monitor por campo inferiores a los límites (246.8 y 245.8 UM para el Campo 1 y 2, respectivamente).
- Coordenadas del isocentro  $(X, Y, Z) = (0, -3.5, -2.5)$  con respecto al sistema de coordenadas definido por los marcadores radioopacos o PTs.

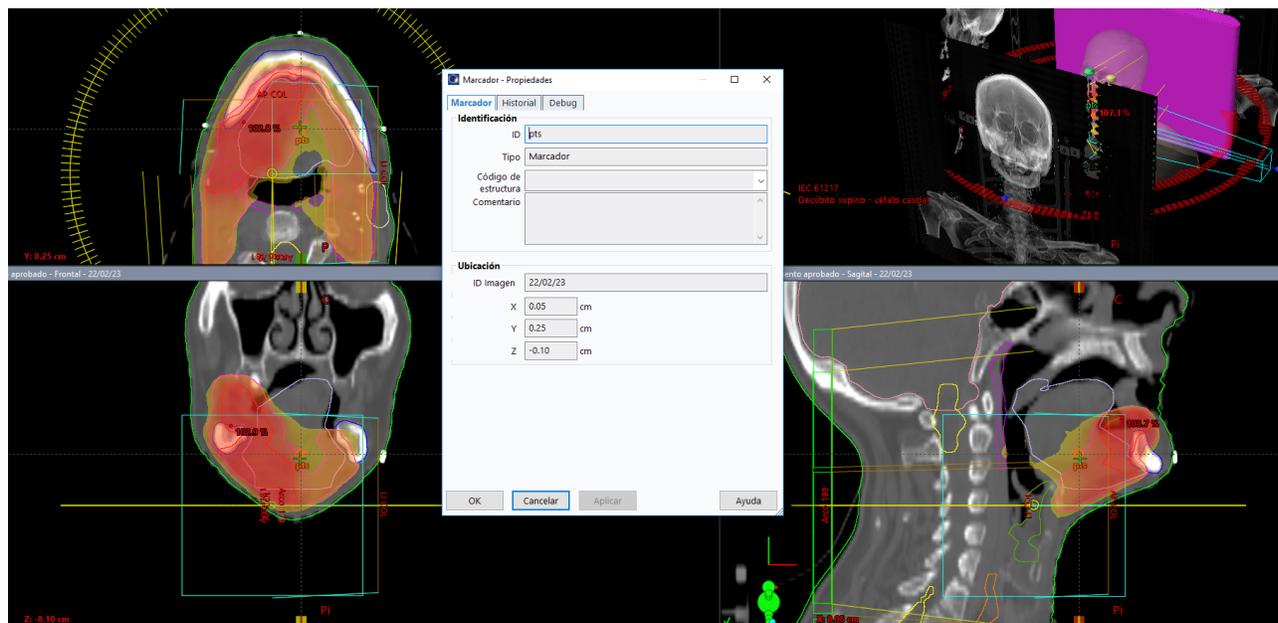


Figura 3.7: Posición del origen de coordenadas definido por los marcadores (puntos blancos) con respecto al de los láseres del LINAC y posición de los marcadores para el paciente A.

- 3 marcadores radioopacos (Figura 3.7) definiendo el origen de coordenadas en  $(X, Y, Z) = (0.05, 0.25, -0.10)$  con respecto al origen definido por los láseres del LINAC.
- El histograma dosis-volumen se muestra en la Figura 3.8.

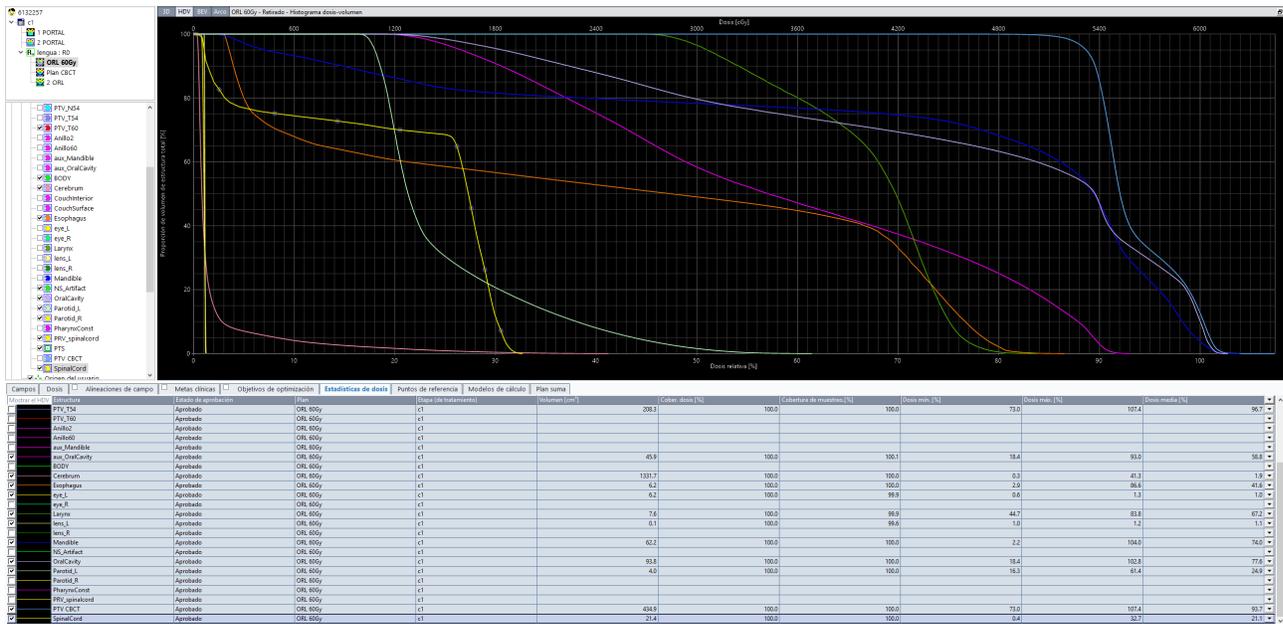


Figura 3.8: Histograma dosis-volumen de los órganos de riesgo del paciente A. En el panel inferior, en la parte izquierda, se puede observar el color que corresponde a cada órgano.

### Paciente B.

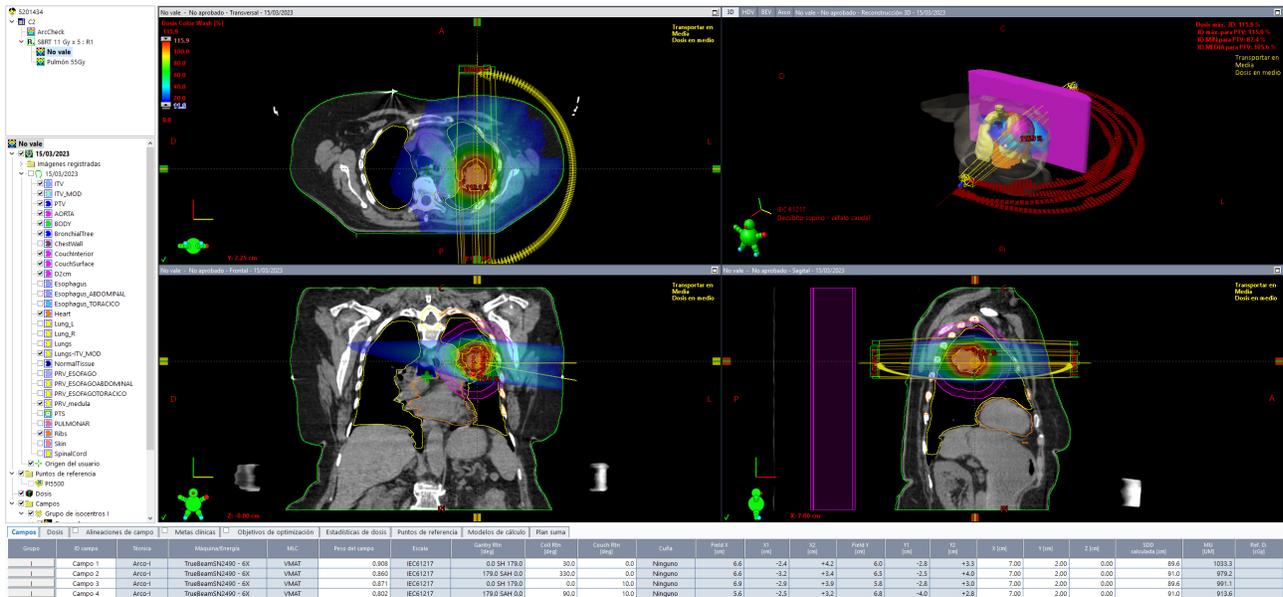


Figura 3.9: Captura de pantalla del planificador para el paciente B.

El plan de tratamiento del paciente B (Figura 3.9) se caracteriza por:

- Únicamente consta de un curso de tratamiento.

- Tumor de pulmón.
- Estructuras contorneadas (cuerpo, esófago, corazón, costillas, médula espinal, etc.).
- Dispone de las estructuras *CouchInterior* y *CouchSurface*, que conforman la camilla de tratamiento del paciente.

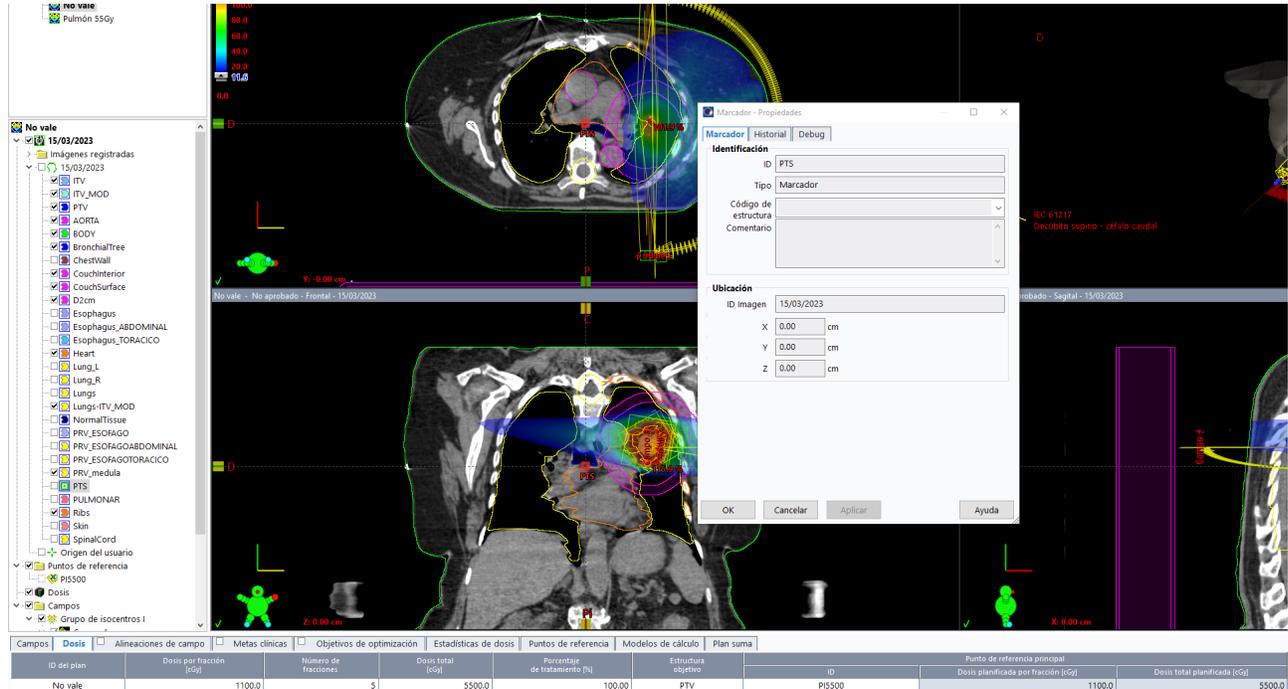


Figura 3.10: Posición del origen de coordenadas definido por los marcadores con respecto al definido por los láseres del LINAC y posición de los marcadores para el paciente B. En la parte inferior se muestra la dosis total, dosis por sesión y el número de sesiones.

- Prescripción de dosis (Figura 3.10): el tumor debe recibir 5500 cGy en 5 sesiones de 1100 cGy.
- Acelerador: TrueBeamSN2490.
- Técnica de tratamiento: SBRT-VMAT (cuatro campos de irradiación).
- Dos arcos con mesa de tratamiento sin rotación y los otros dos con rotación de mesa a  $10^\circ$ . Existe posibilidad de colisión.
- Unidades de monitor por campo inferiores a los límites (1033.3, 979.2, 991.1 y 913.6 UM para los Campos 1, 2, 3 y 4, respectivamente).
- Coordenadas del isocentro  $(X, Y, Z) = (7.00, 2.00, 0.00)$  con respecto al sistema de coordenadas definido por los marcadores.

- 3 marcadores radioopacos (Figura 3.10) definiendo el origen de coordenadas en (X, Y, Z) = (0, 0, 0) con respecto al origen definido por los láseres del LINAC (los marcadores están colocados justo encima de los láseres).
- El histograma dosis-volumen se muestra en la Figura 3.11.

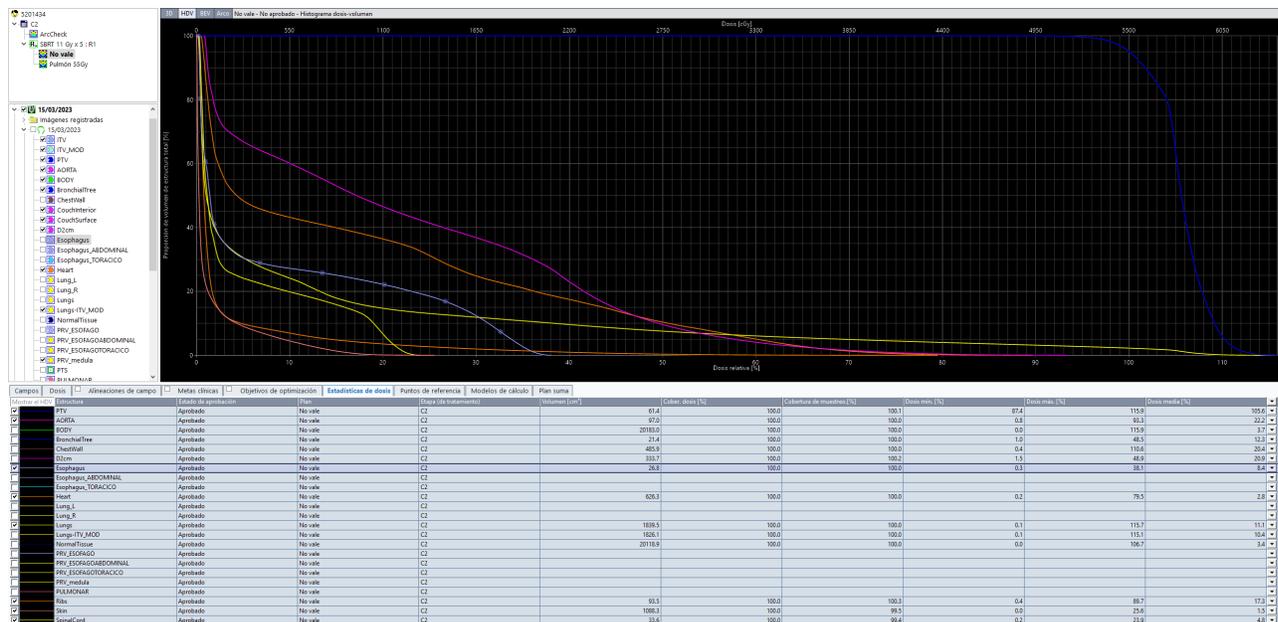


Figura 3.11: Histograma dosis-volumen de los órganos de riesgo del paciente B.

Paciente C.



Figura 3.12: Etapas clínicas y captura de pantalla del planificador para el paciente C.

El plan de tratamiento del paciente C (Figura 3.12) se caracteriza por:

- Consta de tres cursos de tratamiento. El plan de tratamiento se planifica sobre las imágenes del primer curso de tratamiento (más antiguo) en vez en las imágenes TC más actuales.
- Tumor en las vértebras.
- Estructuras contorneadas (cuerpo, CTV y PTV). El histograma dosis-volumen carece de relevancia ya que no hay órganos de riesgo que analizar.
- Dispone de las estructuras *CouchInterior* y *CouchSurface*, que conforman la camilla de tratamiento del paciente.

Campos		<input type="checkbox"/> Dosis	<input type="checkbox"/> Alineaciones de campo	<input type="checkbox"/> Metas clínicas	<input type="checkbox"/> Objetivos de optimización	Estadísticas de dosis	Puntos de referencia	Modelo
ID del plan	Dosis por fracción [cGy]	Número de fracciones	Dosis total [cGy]	Porcentaje de tratamiento [%]				
T7_8Gy	800.0	1	800.0	100.00				

Figura 3.13: Dosis total, dosis por sesión y número de sesiones para el Paciente C.

- Prescripción de dosis (Figura 3.13): el tumor debe recibir 800 cGy en una única sesión.

- Acelerador: TrueBeamSN2490.
- Técnica de tratamiento: Dosis dinámica (3 campos con angulaciones  $220^{\circ}$ ,  $140^{\circ}$  y  $180^{\circ}$ ).
- Mesa de tratamiento sin rotación.
- Unidades de monitor por campo inferiores a los límites (589.1, 567.8 y 476.3 UM para los Campos 1, 2 y 3, respectivamente).
- Coordenadas del isocentro  $(X, Y, Z) = (0.87, 9.38, -12.43)$  con respecto al sistema de coordenadas definido por los marcadores.
- Todos los campos están diseñados para administrarse en el mismo acelerador y tienen el mismo isocentro.

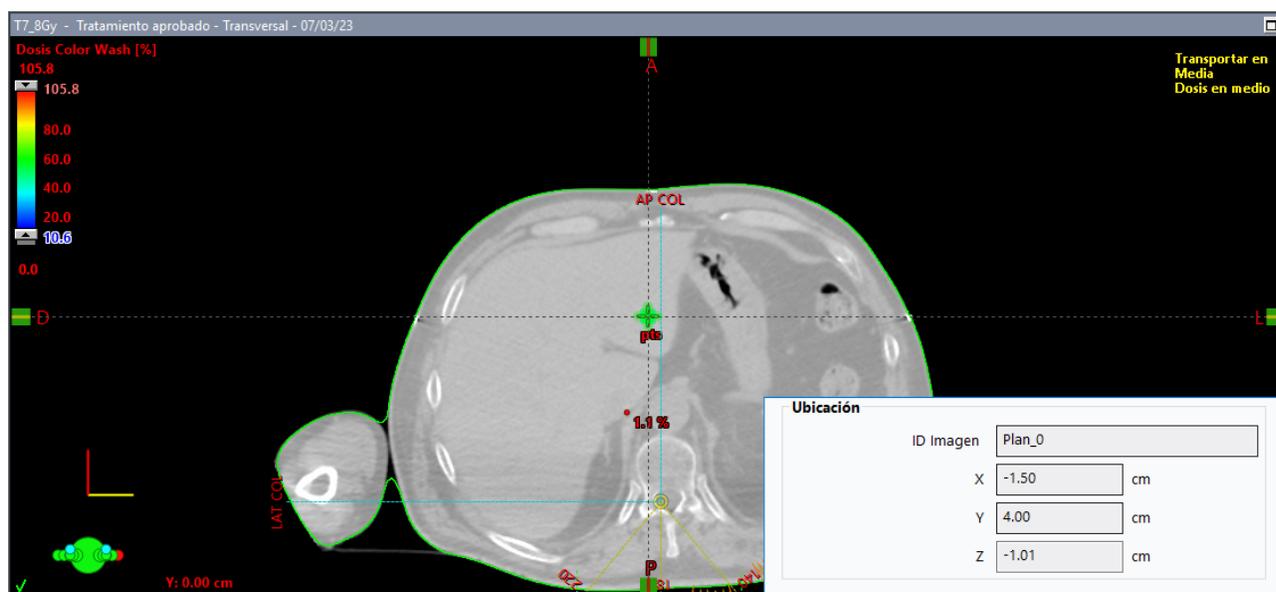


Figura 3.14: Posición del origen de coordenadas definido por los marcadores con respecto al definido por los láseres del LINAC y posición de los marcadores para el paciente C.

- 3 marcadores radioopacos (Figura 3.14) definiendo el origen de coordenadas en  $(X, Y, Z) = (-1.50, 4.00, -1.01)$  con respecto al origen definido por los láseres del LINAC, por lo que los marcadores no están próximos a los láseres.

### 3.7. Ahorro de tiempo en la etapa de verificación del tratamiento

El objetivo de este apartado es estimar cuánto tiempo se ahorra un radiofísico en la etapa de verificación de los tratamiento gracias al *script* diseñado para automatizar el check-list del SOR del HUMV.

El tiempo tomado depende en gran medida del número de estructuras presentes, por lo que se han creado tres grupos: cabeza/cuello, tórax/pecho y abdomen/pelvis/extremidades. De acuerdo con los radiofísicos del Servicio, el tiempo tomado en realizar la check-list para tumores ubicados en la cabeza y cuello es muy superior al resto, donde el tiempo es similar, por lo que el objetivo es comprobar que en el primer grupo es donde más tiempo se ahorra. Para ello, se recogen los tiempos que diez radiofísicos requieren para verificar los pasos de la check-list de forma manual y haciendo uso del *script* para una planificación de cada uno de estas patologías.

# Capítulo 4

## Resultados y análisis

El *script* “Check-list verificación tratamientos SOR”, en lenguaje de programación C#, que se ha implementado para realizar la comprobación de los planes de tratamiento del SOR del HUMV se muestra en el Anexo I. También puede ser encontrado en el repositorio de GitHub [37] o haciendo click [aquí](#). Este script se ejecuta desde la sesión de Eclipse, aplicándose únicamente al paciente cargado en dicho momento.

### 4.1. Funciones del código

Las funciones implementadas comprueban ciertos parámetros del plan de tratamiento y devuelven una cadena de texto que muestra información al usuario sobre los mismos. En este apartado se explica en qué consisten las funciones utilizadas para comprobar los pasos del check-list del SOR, que se muestran en el Anexo I. Desde un ordenador, puede accederse a cada una de ellas haciendo click sobre cada paso en azul.

#### Paso 1. Consultar notas.

- Función **CheckNotas**. Esta función devuelve los posibles comentarios que algún profesional del SOR haya escrito en relación al plan de tratamiento que se está revisando. Para acceder a las notas escritas se utiliza la clase *PlanSetup*, dentro de la cual se encuentra la clase *Course*, que contiene el valor *Comment*, de tipo string<sup>1</sup>. La función tiene como argumento la primera clase y devuelve el valor en un mensaje.

**Paso 2. Abrir láminas.** Dado que este paso únicamente se aplica a planes de tratamiento de tumores de mama, no se ha implementado una función específica. En pantalla aparece un mensaje de alerta: “2) Abrir láminas: revisar en caso de mama”.

**Paso 3. Correspondencia del nombre del paciente, prescripción y etapa de tratamiento.** Se han conectado dos funciones: **CheckCT**, para comprobar el curso de tratamiento

---

<sup>1</sup>En cualquier lenguaje de programación, un string es una secuencia de caracteres usado para representar el texto.

y **CheckName**, que devuelve el nombre del paciente.

- Función **CheckCT**: comprueba que el TC sobre el que se ha calculado el plan de tratamiento es el adecuado y que, por ejemplo, no corresponde con TCs anteriores de otros tratamientos previos. Tiene como argumento la clase *ScriptContext*. Esta función accede a las fechas de realización de las imágenes de TC. Si sólo cuenta con un TC, devuelve un mensaje indicando que estamos en la etapa de tratamiento adecuada. En cambio si dispone de varios TCs, indica si el TC sobre el que se ha planificado es el más reciente o no.
- Función **CheckName**: Esta función devuelve el nombre y apellidos del paciente, cuyos datos se encuentran mediante la clase *StructureSet*.

**Paso 4. Verificación del origen PTs.** Se han diseñado dos funciones.

- Función **CheckMarkers**: comprueba que existan los tres marcadores radioopacos colocados sobre el paciente durante la simulación TC y estén alineados con los ejes del origen definido por el LINAC. Para comprobar esto, se define para cada uno de los marcadores una matriz de píxeles de 4mm×4mm con el marcador en el centro, sobre el corte del TC en el que se encuentran situados. Dicha matriz debe contener al menos un píxel con densidad mayor que 1500 UH (unidades Hounsfield<sup>2</sup> [38]).
- Función **CheckOrigin**: verifica que el origen definido por los marcadores radioopacos se corresponde con el punto (0, 0, 0) con respecto al origen definido por el LINAC, de tal manera que ambos orígenes coinciden. Este valor se obtiene mediante el atributo *Image.UserOrigin*, dentro de la clase *ScriptContext*.

**Paso 5. Verificación de la dosis de prescripción y del número de sesiones correcto.**

- Función **CheckPrescription**: comprueba que la dosis recibida en el punto de referencia de dosis se corresponde con la prescrita. El punto de referencia de dosis es un punto físico del paciente virtual que se usa para almacenar las dosis total, diaria y por sesión del tumor (PTV). En primer lugar, comprueba que exista dicho punto de dosis y que el plan disponga de una prescripción de dosis. A continuación, busca la estructura u órgano de mayor dosis, y comprueba que dicho valor corresponde con la dosis del punto de referencia, lo cual significa que dicha estructura es el tumor.

Esta función también comprueba que el número de sesiones introducido en el plan coincide con el prescrito. Para ello accede al atributo *NumberOfFractions* dentro de la subclase *RTPrescriptionTarget*.

**Paso 6. Correcta posición del paciente y mesa de tratamiento.** Para comprobar este paso se diseñan dos funciones: **CheckCouch**, que comprueba que existen las estructuras

---

<sup>2</sup>La escala de Unidades Hounsfield es una escala cuantitativa utilizada en los estudios de tomografía computarizada (TC) para describir los diferentes niveles de radiodensidad (capacidad para atenuar o disminuir la intensidad de una radiación) de los tejidos humanos.

*CouchInterior* y *CouchSurface* y **CheckMaquinaMesa**, que otorga la ID y modelo de la máquina.

- Función **CheckCouch**: comprueba que en el apartado izquierdo de estructuras existen los contornos “CouchInterior” y “CouchSurface”, que constituyen la camilla de tratamiento, y sean únicos. Utiliza como argumento las clases *StructureSet* y *PlanSetup*.
- Función **CheckMaquinaMesa**: en caso de que exista la estructura *CouchInterior*, devuelve el ID y el modelo de la máquina haciendo uso de la clase *Structure* dentro de la clase *StructureSet*.

**Paso 7. Técnica de tratamiento adecuada según la prescripción y nombre correcto de los campos.**

- Función **CheckVMATorIMRT**: comprueba si la técnica es VMAT o IMRT. Si la técnica es la VMAT, los campos se denominan Campo o Arco 1, 2, 3, etc. Es decir, deben contener las palabras “Campo” o “Arco”. Por otro lado, si la técnica es IMRT, la nomenclatura del campo corresponde con el ángulo de rotación del grantry desde el cual se irradia (sin los grados), por ejemplo, 90, 270, 65, etc. Si la técnica es IMRT y contiene campos nombrados con la palabra “Campo” o “Arco”, estarían mal nombrados. Si la técnica utilizada no es IMRT o VMAT, la función devuelve un mensaje de error. Utiliza el atributo *MLCPlanType*, que se encuentra en el interior de la subclase *Beams*. Como argumento se introduce la clase principal *PlanSetup*.

**Paso 8. Posición correcta del isocentro.** Se verifica en la función del paso 9.

**Paso 9. Todos los campos asociados al mismo acelerador y con el mismo isocentro.**

- Función **CheckMismoAcelerador**: comprueba que todos los campos están referidos al mismo acelerador. Su funcionamiento es el siguiente: se crea una variante booleana<sup>3</sup>, inicialmente “True”, y se obtiene el ID del acelerador para cada campo, si en algún momento los IDs no coinciden, esta variable booleana cambia a “False” e interrumpe la comprobación de los campos. Si al final de la función la variante booleana es de tipo “True”, las máquinas de cada campo han coincidido y si es de tipo “False” se habrán detectado diferentes aceleradores para un mismo plan. Utiliza el atributo *TreatmentUnit.Id*, que se encuentra en el interior de la subclase *Beams*.
- Función **CheckMismoIsocentro**: trabaja de forma similar a la función anterior. La principal diferencia es que el isocentro está compuesto por tres coordenadas (x, y, z), por lo que debe repetirse el proceso una vez para cada una. Puesto que en el octavo paso se pide la posición del isocentro, esta función, además de verificar si las coordenadas del isocentro de todos los campos coinciden o no, devuelve las coordenadas del isocentro (en caso de coincidir), indicando si están o no redondeadas, lo cual es fundamental para facilitar el posicionamiento del paciente en el acelerador. Utiliza el atributo *IsocenterPosition*, que

<sup>3</sup>Los números enteros pueden ser 1, 2, 3, etc. mientras que los datos de tipo booleano sólo pueden tener los valores “True” o “False”.

se encuentra en el interior de la subclase *Beams*.

**Paso 10. Matriz de dosis correcta.** La matriz de dosis<sup>4</sup> no es un valor al que se pueda acceder desde la biblioteca de funciones de la API de Eclipse, por lo que inevitablemente es un paso que debe ser comprobado manualmente. Generalmente, la resolución de la malla de cálculo es de 2,5 mm salvo excepciones como en tratamientos de SBRT, en la cual se utiliza una resolución de 1.25 mm. Así pues, se ha decidido mostrar un mensaje donde se recuerde que la comprobación debe realizarse por el usuario.

#### Paso 11. Verificación del algoritmo de cálculo.

- Función **CheckCalculationModel**: lee el algoritmo de cálculo elegido para el cálculo de la distribución de la dosis del plan y muestra un mensaje de error en caso de no encontrar las etiquetas “Acuros”, “Acurous” o “AAA”. Esta función tiene como argumento la clase *PlanSetup*, dentro de la cual se encuentra el atributo *PhotonCalculationModel*.

#### Paso 12. Verificación de que no hay colisiones con la mesa, los soportes y/o el paciente.

- Función **CheckColisiones**: en determinadas angulaciones el gantry del acelerador puede colisionar con la mesa, los soportes de inmovilización o el paciente. Cuando la mesa presenta una angulación, es decir, la rotación de la mesa (*CouchRTN*) es no nula, la probabilidad de colisión es mayor por una cuestión de geometría. Por tanto, está función lee el valor de *CouchRTN* para cada campo y si en todos los casos es cero, devuelve un mensaje informando de que la probabilidad de colisión es ínfima. En caso de que *CouchRTN* sea menor que tres grados de rotación, la probabilidad sigue siendo baja pero no nula. Finalmente, si *CouchRTN* es superior a tres grados, se devuelve un mensaje de advertencia ante posibles colisiones. El ángulo de rotación de la mesa se corresponde con el valor *PatientSupportAngle*, dentro del atributo *ControlPoints*.

#### Paso 13. Verificación de que se cumplen los objetivos y restricciones.

- Función **CheckConstraints**: se quiere crear un diccionario de posibles nombres de estructuras que intenta contemplar las nomenclaturas de los órganos contorneados del plan. Para ello, se crea un diccionario en el que la etiqueta corresponde a la estructura y el valor se corresponde con los nombres de una lista de expresiones regulares [39] que intentan coincidir con los órganos de la prescripción. La etiqueta es un número que se corresponde con el ID del Foundational Model of Anatomy [40], que es el usado por Eclipse. Por ejemplo, la etiqueta 7647 corresponde a la médula y acepta los valores “Med” y “Spinal Cord”, así como todas sus variantes en cuanto a espacios o mayúsculas. Si se hubiese escrito como “Medlua” sería necesario añadir esta al diccionario.

A continuación, se identifican los órganos de riesgo, tal y como aparecen en la prescripción, con aquellos que aparecen como estructuras, haciendo uso del diccionario creado anteriormente. También se ha tenido en cuenta que en la prescripción los órganos pueden

<sup>4</sup>Representación tridimensional de la distribución de la radiación en el tejido.

aparecer como Contralateral o Ipsilateral y no como Izquierda o Derecha, por lo que se ha hecho equivalente contra/ipsi a izquierda/derecha.

Finalmente, se identifican los índices de limitación de dosis para cada órgano (dosis media, dosis máxima, volumen permitido para una dosis dada o viceversa). Una vez identificado el índice de limitación de dosis, se comprueba en el histograma dosis-volumen (DVH) los índices dosis-volumen conseguidos en el plan de tratamiento diseñado y se comparan con los prescritos. La función devuelve un informe con dichos índices dosis-volumen y posibles problemas.

Esta función tiene como argumento las clases *PlanSetup*, *StructureSet*, *Image* y *RTPrescription* y de ellas se utilizan una gran cantidad de atributos como pueden ser *structure.Id* para obtener los nombres de las estructuras o *ps.GetDoseAtVolume* para obtener la dosis en un determinado volumen.

**Paso 14. La distribución de dosis correcta por plano, por plan y por el tratamiento completo.** Dada la complejidad de programación de esta función, se decide únicamente devolver un mensaje de advertencia de que este paso debe realizarlo el usuario mediante una inspección visual del plan.

**Paso 15. Verificación de que no se exceden los límites de UM de la máquina.**

- Función **CheckUMcGy**: lee el valor de las unidades monitor (UM) administradas por cada campo y si alguno de ellos es superior a 1500 UH, el bucle se interrumpe y se informa al usuario de la necesidad de dividir este campo en dos, dividiendo de igual forma las UHs. Esta función utiliza el valor *Value* dentro del atributo *Meterset*.

**Paso 16. Colocación correcta de Bolus: posición, densidad, espesor y asignación de haces.**

- Función **CheckBolus**. Esta función recorre cada campo y comprueba que el espesor de los bolus coincide en cada uno de ellos (y por tanto, todos los campos tienen asociados el bolus). Si los espesores coinciden porque son nulos, es que no hay bolus en ningún campo. Si los espesores coinciden y no son nulos, la función devuelve un mensaje indicando que todos los campos cuentan con bolus cuyos espesores coinciden y da el valor correspondiente. Esta función utiliza principalmente el atributo *BolusThickness* en el interior de la clase *RTPrescription*.

**Paso (adicional) 17. Comprobación de que no hay estructuras vacías y existe una estructura “BODY” de tipo “EXTERNAL”.** Una estructura llamada body de tipo externo significa que todo el cuerpo del paciente ha sido contorneado, por lo que los límites del mismo se encuentran establecidos. Este paso lo realizan los oncólogos radioterápicos por lo menos una vez por planificación por lo que, aunque no se pida, se ha realizado un paso que puede resultar de ayuda al personal del servicio. El motivo por el que este paso resulta útil es porque una estructura vacía o una estructura “body” que no defina un contorno pueden dar lugar a errores a la hora de hacer la planificación.

- Función **CheckStructures**. Esta función recorre cada estructura del planificador y, en caso de estar alguna vacía, interrumpe el bucle y devuelve el nombre de la estructura vacía. Una vez que no hay estructuras vacías, busca la estructura “BODY”. Si esta no existe o existe pero no es un contorno del cuerpo, alerta al radiofísico. En caso contrario, confirma que todo es correcto.

## 4.2. Ejecución del código

A continuación se van a mostrar las ventanas emergentes tras ejecutar el *script* “Check-list verificación tratamientos SOR” en los pacientes “A”, “B” y “C”.

- **Paciente A.**

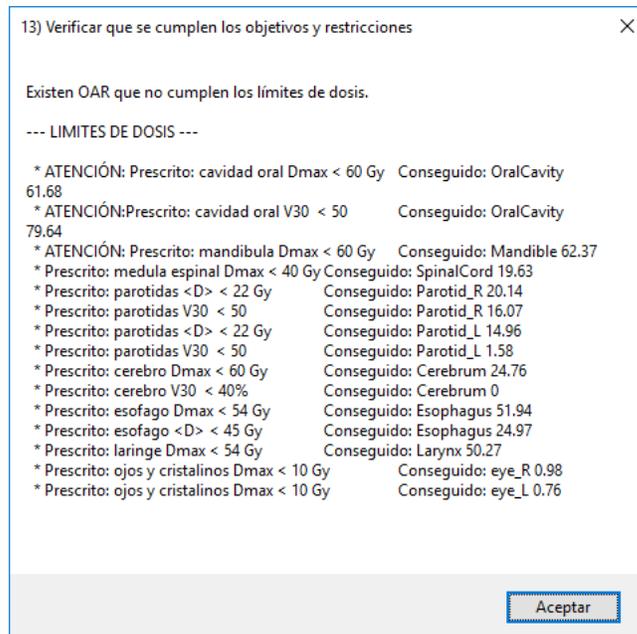
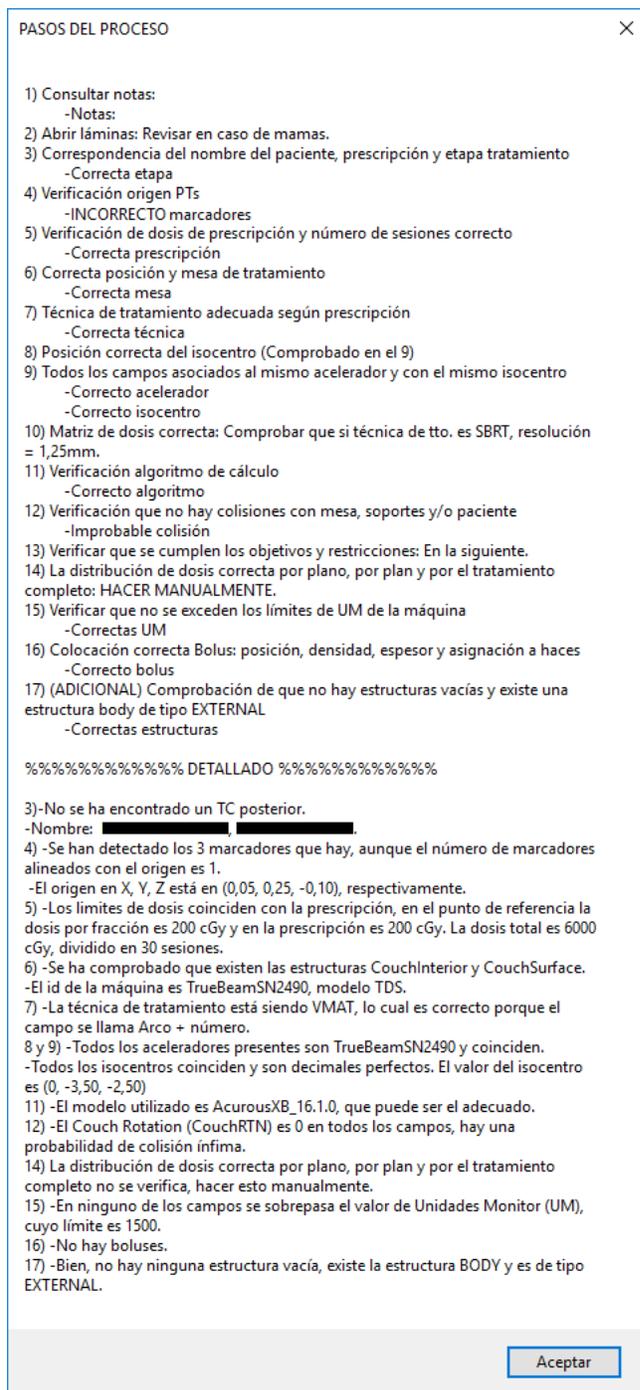


Figura 4.1: Ventanas emergentes al ejecutar el código “Check-list verificación tratamientos SOR” para el paciente A.

■ Paciente B.

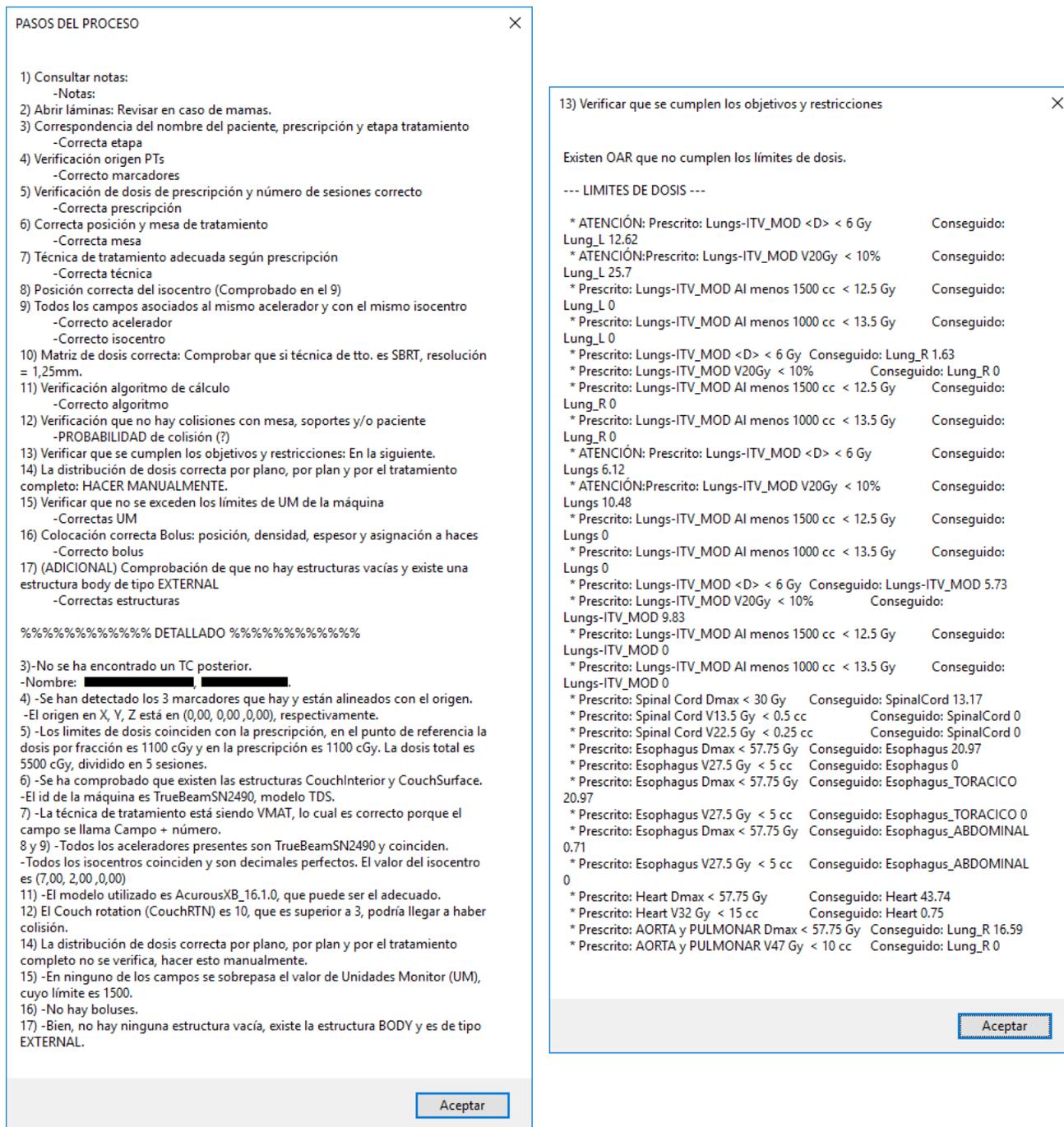


Figura 4.2: Ventanas emergentes al ejecutar el código “Check-list verificación tratamientos SOR” para el paciente B.

## ■ Paciente C.

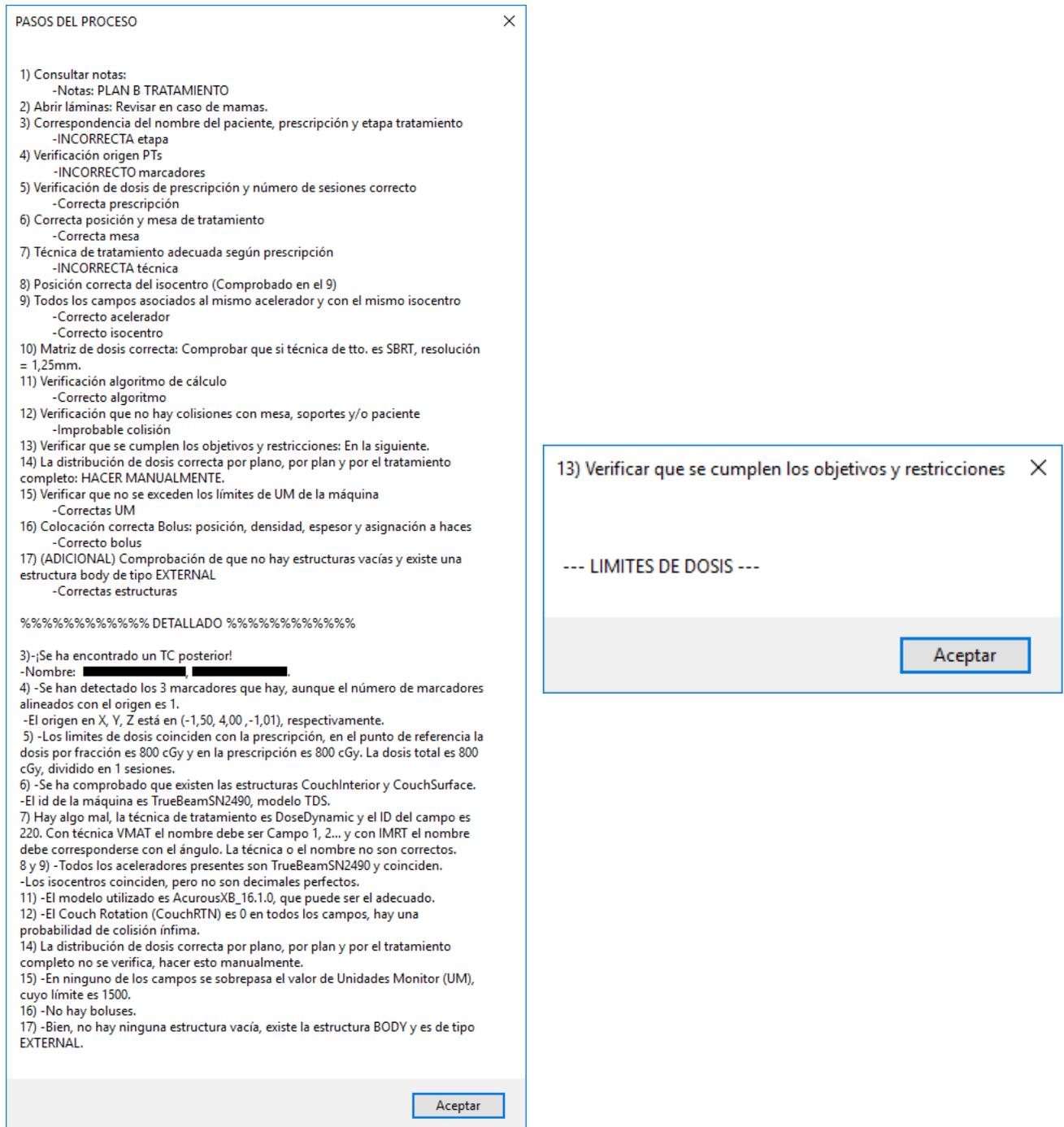


Figura 4.3: Ventanas emergentes al ejecutar el código “Check-list verificación tratamientos SOR” para el paciente C.

### 4.3. Ahorro de tiempo en la etapa de verificación del tratamiento

A continuación, se muestran los tiempos en minutos (por comodidad, los indicamos aquí con la letra m, teniendo en cuenta que no hay posible confusión con metros) que requieren diez radiofísicos para realizar el check-list del SOR de forma manual y haciendo uso del *script* para tres posibles patologías:

<b>Sin usar código</b>	1	2	3	4	5	6	7	8	9	10	Media (m)	Prob.
Cab./Cue. (m)	-	25	20	28	31	22	-	30	27	21	$26 \pm 4$	0.2
Tór./Pec. (m)	12	-	14	10	17	13	11	15	9	11	$11 \pm 2$	0.4
Ab./Pel./Ex. (m)	9	11	12	-	10	8	7	12	6	10	$9 \pm 2$	0.4
<b>MEDIA (m)</b>											<b><math>13.2 \pm 1.4</math></b>	

<b>Usando código</b>	1	2	3	4	5	6	7	8	9	10	Media (m)	Prob.
Cab./Cue. (m)	-	12	10	13	13	12	-	14	14	12	$12.5 \pm 1.2$	0.2
Tór./Pec. (m)	9	-	11	10	12	11	10	12	9	10	$10.4 \pm 1.1$	0.4
Ab./Pel./Ex. (m)	9	10	12	-	9	8	8	11	7	9	$9.2 \pm 1.5$	0.4
<b>MEDIA (m)</b>											<b><math>10.3 \pm 0.8</math></b>	

Tabla 4.1: Tiempos (minutos) que requieren diez radiofísicos para verificar la check-list de verificación de tratamientos del SOR para diferentes patologías de forma manual (tabla superior) y haciendo uso del código (tabla inferior), respectivamente. Las casillas donde aparece un guión se corresponde con no haber encontrado tratamientos para esa parte del cuerpo en el historial de pacientes del radiofísico. “Prob” se corresponde con la probabilidad de que el tumor se encuentre en esa ubicación. El método de estimación de errores puede ser visto en el apartado “Anexo II: Estimación de errores”.

# Capítulo 5

## Discusión

Se ha diseñado un *script* para automatizar la revisión de los requisitos recogidos en la check-list de verificación de los tratamientos del SOR. Este *script* permite ahorrar tiempo en una de las tareas recurrentes en el día a día de un radiofísico hospitalario y disminuir las fuentes de error humano. El *script* no está exento de posibles fallos por lo que siempre tendrá que haber una segunda verificación hasta que haya sido comprobado lo bastante como para asegurar su validez.

Asimismo, es usual que después de la verificación de la planificación vengan etapas posteriores como la aceptación de la planificación por parte del oncólogo radioterápico, la verificación pretratamiento en el acelerador, etc., por lo que se realiza una segunda verificación al final del proceso por otro radiofísico y aunque se suelen revisar parámetros diferentes en ambas verificaciones, acaba habiendo redundancias, con la resultante pérdida de tiempo. Gracias a este tipo de scripts, es posible mantener dicha redundancia con escasa pérdida de tiempo.

Por otro lado, la creación de *scripts* en modo *plug-in scripts* aporta ventajas como la facilidad en la edición y la rápida ejecución sobre un paciente particular. Para ejecutar el *script*, únicamente se necesita el código de texto guardado con extensión .cs, el cual se abre con la herramienta scripts de Eclipse. Al igual que en este trabajo, otros profesionales han optado por esta opción, como el *script* diseñado en el Hospital Clínico Universitario Lozano Blesa [41]. Sin embargo, se debe tener en cuenta que la opción *standalone executable script* se basa en que el usuario final tiene que lanzar un programa .exe que le da acceso a cualquier paciente de la base de datos de Eclipse una vez lo identifica a través de su número de identificación (ID). Este tipo de *script* es fácilmente depurable y en caso necesario la conversión al tipo textitplug-in script es trivial [41].

Tras realizar diferentes versiones del *script*, se ha decidido presentar los resultados del check-list en modo reducido, donde ágilmente seamos capaces de detectar qué paso ha resultado incorrecto. Seguidamente, el *script* muestra los mismos resultados pero más detallados, con información particular sobre cada prueba y explicando, en caso de encontrar algo incorrecto, la fuente de error.

Todos los pasos a comprobar del check-list del SOR han sido implementados en el *script*, excepto los pasos 2, 10 y 14. El no haber conseguido realizar el **paso 2** (Abrir láminas) no es un problema grave, ya que es únicamente relevante en el caso de que se esté tratando una mama (1 de cada 8 tratamientos [42]). Los planes de cabeza y cuello y SBRT, que eran nuestro objetivo, no se ven afectados. El **paso 10** (Matriz de dosis correcta) es especialmente importante para SBRT, ya que debe tener una resolución de cálculo de 1.25 mm, por lo que es beneficioso el mensaje de alerta emergente recordando verificar esto manualmente. Para el resto de de tratamientos la resolución es siempre 2.5 mm. Para el **paso 14** (La distribución de dosis correcta por plano, por plan y por el tratamiento completo) podría diseñarse un código que hiciese un estudio de las distribuciones de dosis para cada corte del volumen a tratar. El diseño de esto sería muy complejo pero podría ser implementado en trabajos futuros.

La automatización del análisis dosimétrico de los histogramas dosis-volumen es especialmente importante en aquellos tratamientos con un gran número de estructuras contorneadas, lo cual convierte en una tarea tediosa y repetitiva esta labor. Es por esto, que se decide elegir ejemplos de pacientes de cabeza y cuello (paciente A) y una SBRT de pulmón (paciente B). Además, el paciente C creado nos ha servido para realizar el “control de calidad” del *script* ya que en él se puede comprobar que los pasos incorrectos también son detectados.

La ejecución del *script* en el paciente A nos ha alertado de que las coordenadas del origen (0.05, 0.25, -0.10), determinadas por las proyecciones de los PTs, son incorrectas. Sin embargo, estas coordenadas son muy próximas a cero. Esto significa que cuando los técnicos de máquina posicionan al paciente con la ayuda de los láseres en el punto indicado por los marcadores radioopacos y lo desplazan (verticalmente, longitudinalmente y lateralmente) en función de las coordenadas del isocentro del plan, el isocentro del acelerador quedará muy próximo al esperado ya que las coordenadas del isocentro no varían apreciablemente a causa del error.

Por otro lado, tanto el PTV como la mayoría de los órganos de riesgo cumplen la prescripción y las restricciones de dosis programadas (12 de los 15 casos) y por tanto el resultado es correcto. Sin embargo, en la cavidad oral y la mandíbula se detectan máximos de dosis superiores a la dosis de prescripción (60 Gy), por lo que el *script* nos alerta. Es decisión del radiofísico hospitalario y del oncólogo radioterápico discutir si esto es aceptable teniendo en cuenta las dificultades del caso, la anatomía del paciente y la localización del tumor. Asimismo, el *script* da cuenta de que la cavidad oral sobrepasa la dosis máxima permitida en aproximadamente 50 Gy, lo cual es irreal y se considera que es un error de lectura por parte del *script*. En definitiva, en catorce de los quince órganos de riesgo (93 %) se han revisado las condiciones dosimétricas correctamente, lo que supone un ahorro muy significativo del tiempo. De estos catorce órganos, únicamente dos de ellos no cumplen las limitaciones de dosis, sobrepasándose una dosis inferior al 4 %.

La ejecución del *script* en el paciente B nos ha notificado que existe cierta probabilidad de colisión debido a que la rotación de la camilla es de  $10^\circ$ . Esto es muy útil ya que permite al radiofísico hospitalario recordar la importancia de realizar simulaciones en el acelerador, haciendo uso de los inmovilizadores del paciente y los ángulos de gantry y mesa pertinentes, para detectar la posible existencia o no de colisiones. En ocasiones la cabeza del gantry choca

con la mesa, o también, con los inmovilizadores del paciente o su propia anatomía (brazos, piernas, etc.).

En el caso del paciente B, destaca la importancia del *script* ya que cuenta con 29 limitaciones de dosis que deben ser comprobadas, de las cuales 27 de ellas han sido revisadas correctamente (93%) y únicamente dos no cumplen las restricciones dosimétricas.

El paciente C se ha creado para comprobar la eficacia del *script*. En primer lugar, nos da aviso de la existencia de un TC posterior. Este error puede ocurrir cuando se trata de un paciente que tiene varios cursos de tratamiento poco separados en el tiempo y cuyos PTVs están próximamente localizados entre sí. Esto significa que las imágenes TC de los distintos cursos de tratamiento van a ser similares entre sí, pero lo suficientemente diferentes como para que la distribución de dosis que se planifica difiera de la que se administraría en la realidad. Si el primer tratamiento de un paciente es un tumor de pulmón y posteriormente se le tratan unas lesiones cerebrales, la probabilidad de confundir las imágenes TC es mínima. Sin embargo, si el paciente se ha tratado de una mama y posteriormente requiere un nuevo tratamiento en la mama contralateral, puede existir cierta probabilidad de confundir las imágenes o seleccionar imágenes obsoletas y por tanto administrar una dosis inadecuada al paciente.

En cuanto a la técnica, “Dynamic Dose” es equivalente a IMRT, pero el código contempla IMRT o VMAT por lo que notifica de un error. Esta técnica podría ser fácilmente añadida como una posibilidad más, sin embargo, es preferible un código que compruebe los aspectos más comunes, de tal forma que si en algún paso se tiene algo infrecuente, el experto puede ver de forma explícita que es así. Por otro lado, en cuanto a la posición del isocentro, el *script* detecta que las coordenadas del mismo son (-0,87, -9,38, -12,43) y no están redondeadas. Esto no implica error en la deposición de la dosis del paciente, sin embargo, que estén redondeadas facilita la labor de posicionamiento del paciente en la camilla del acelerador a los técnicos de radioterapia.

En este caso, por ser un tratamiento paliativo de una única sesión, los órganos de riesgo no se contornean por lo que el paso de comprobación de las restricciones y limitaciones de dosis ha quedado vacía.

Por último, se ha estimado el ahorro de tiempo de un radiofísico hospitalario si realiza la verificación de los tratamiento mediante el *script* diseñado respecto a una comprobación manual. Fundamentalmente, este ahorro se debe a la automatización de la etapa de revisión de las restricciones dosimétricas mediante el análisis de los HDVs. Este paso es especialmente tedioso en planificaciones que cuentan con un gran número de estructuras y limitaciones de dosis, por lo que el beneficio del *script* es más notable. En los tratamientos de cabeza y cuello el tiempo empleado en comprobar el check-list disminuye de  $(26 \pm 4)$  minutos a  $(12,5 \pm 1,2)$  gracias al *script* por lo que se tiene un ahorro de tiempo del 50%, aproximadamente. Por lo que se ha dicho, podría pensarse que se está asumiendo que el *script* sólo va a ser utilizado para la check-list cuando no es así. Los límites de dosis se comprueban constantemente también en la etapa de planificación y en esos casos el código también puede ser utilizado.

La automatización de la verificación de tratamientos del SOR podría extrapolarse a todo tipo

de planificaciones de tratamiento (mama, próstata, vulva, pelvis, páncreas, extremidades, etc.) y de técnicas de irradiación con partículas como electrones, protones, etc., siempre y cuando el TPS sea Eclipse.

# Capítulo 6

## Conclusiones y trabajo a futuro

El script “Check-list verificación tratamientos SOR” es capaz de revisar los ítems de la lista de comprobación correspondiente a la etapa de verificación del tratamiento del SOR del HUMV. Su implementación supone un ahorro de tiempo significativo así como la disminución de posibles errores humanos.

Dentro de estos pasos, se destaca la importancia del paso número 13 (Verificación de que se cumplen los objetivos y restricciones de dosis), que en función del número de órganos presentes puede exigir mucho tiempo de revisión. Si el número de órganos presentes es reducido, el *script* es capaz de cubrir la totalidad de este paso, mientras que si es elevado, como se ha visto en dos pacientes sujeto, hay algunas estructuras cuya dosis no se lee correctamente.

A la hora de realizar este trabajo las principales adversidades han sido comprender en profundidad la física de los pasos de la check-list y aprender a programar en el lenguaje C# sin ningún conocimiento inicial. Los conocimientos adquiridos sobre física de partículas a lo largo del Grado en Física han sido fundamentales para comprender la base de la radioterapia. Los conocimientos previos del lenguaje *Python*, han resultado de gran utilidad para el entendimiento de C#.

La automatización de la verificación de tratamientos del SOR podría extrapolarse a todo tipo de planificaciones de tratamiento (mama, próstata, vulva, pelvis, páncreas, extremidades...) y de técnicas de irradiación con partículas como electrones, protones, etc. El *script* se ha diseñado para tratamientos con fotones pero puede extrapolarse fácilmente a tratamientos con electrones o protones como el que se instalará próximamente en el HUMV. En función de la patología, el *script* debe adecuarse para que su ejecución esté optimizada. Además, podría añadirse un código de colores para mejorar la estética de los resultados finales tras la ejecución del *script*.

# Bibliografía

- [1] Epdata: El cáncer en España, datos y estadísticas, consultado en mayo de 2023. [Enlace](#).
- [2] Sociedad española de oncología radioterápica: Historia de la radioterapia, consultado en mayo de 2023. [Enlace](#).
- [3] Instituto nacional del cáncer: Radiación ionizante, consultado en mayo de 2023. [Enlace](#).
- [4] Wikipedia: Radiación ionizante, consultado en mayo de 2023. [Enlace](#).
- [5] Consejo de seguridad nuclear: Interacción de la radiación con la materia, consultado en mayo de 2023. [Enlace](#).
- [6] Kaunas University of Technology. Radiation interaction with condensed matter, consultado en mayo de 2023. [Enlace](#).
- [7] Organismo internacional de energía atómica: Definición de dosis, consultado en mayo de 2023. [Enlace](#).
- [8] Sociedad Española de Física Médica, consultado en mayo de 2023. [Enlace](#).
- [9] Centro de investigaciones energéticas, medioambientales y tecnológicas: Aspectos generales de la interacción de la radiación con el medio biológico, consultado en mayo de 2023. [Enlace](#).
- [10] Justo R. Venereo Gutiérrez. Scielo: Daño oxidativo, radicales libres y antioxidantes, consultado en mayo de 2023. [Enlace](#).
- [11] ScienceDirect. Acción directa e indirecta, consultado en mayo de 2023. [Enlace](#).
- [12] V. Jayalekshmi y T.K. Abhilash G. Uma Devi. Gamma radiation-induced oxidative stress and apoptosis in aortic endothelial cells: Protection by alpha-lipoic acid. *Free Radical Biology and Medicine*, 2012.
- [13] Sravanth H.K. and Annapoorni R. Evidence for a mitochondrial-k<sup>+</sup> channel axis in the cytotoxic action of simian virus 40 small t antigen. *Cancer Research*, 83(17):8565–74, 2009.
- [14] Antonio B.S. y Damián G.L. *Fundamentos de Física Médica Volumen 8: Radiobiología y*

- principios de Oncología*. ADI, 1 edition, 2017.
- [15] Instituto nacional del cáncer. mutación, consultado en mayo de 2023. [Enlace](#).
- [16] B. McClintock y H. J. Muller. La mutación, consultado en mayo de 2023. [Enlace](#).
- [17] Instituto nacional del cáncer. [Página oficial](#).
- [18] Centros para el control y la prevención de enfermedades. tratamientos contra el cáncer, consultado en mayo de 2023. [Enlace](#).
- [19] American cancer society. radioterapia con rayo externo, consultado en mayo de 2023. [Enlace](#).
- [20] Técnicas de radioterapia externa, consultado en mayo de 2023. [Enlace](#).
- [21] M. J. Zelefsky et al. *Stereotactic Body Radiotherapy for Localized Prostate Cancer: Pooled Analysis From a Multi-institutional Consortium of Prospective Phase II Trials*. Vol 109, no. 2, pp. 217-221. Radiotherapy and Oncology, 2013.
- [22] Y. Zhang et al. *Volumetric modulated arc therapy (VMAT) vs. intensity-modulated radiotherapy (IMRT) for nasopharyngeal carcinoma: A dosimetric comparison study*. Vol 43. no. 4, pp. 303-310. Medical Dosimetry, 2019.
- [23] A. J. Ho et al. Stereotactic body radiation therapy for oligometastatic prostate cancer. *International Journal of Radiation Oncology, Biology, Physics*, 99(2):353–360, 2017.
- [24] Carlos Fernández Fernández. Servicio de radiofísica y protección radiológica. centro oncológico de galicia. simulación de tratamientos en radioterapia, consultado en junio de 2023.
- [25] Acelerador lineal, consultado en junio de 2023. [Enlace](#).
- [26] Cristina P.O. y María C.Z.A. Araceli H.V., Antonio B.S. *Fundamentos de Física Médica, Volumen 3: Radioterapia externa I. Bases físicas, equipos, determinación de la dosis absorbida y programa de garantía de calidad*. ADI, 1 edition, 2012.
- [27] y Esther M.C. Antonio B.S. *Fundamentos de Física Médica, Volumen 4: Radioterapia externa II. Dosimetría clínica, algoritmos de cálculo, sistemas de planificación y control de calidad*. ADI, 1 edition, 2013.
- [28] Wikipedia. radioterapia conformada en tres dimensiones., consultado en mayo de 2023. [Enlace](#).
- [29] Research Gate: histograma, consultado en mayo de 2023. [Enlace](#).
- [30] Wikipedia: órgano de riesgo, consultado en mayo de 2023. [Enlace](#).

- 
- [31] Sistema de información oncológica aria, consultado en junio de 2023. [Página oficial](#).
- [32] Instituto zunino. sistemas de planificación, consultado en mayo de 2023. [Enlace](#).
- [33] Eclipse | varian, consultado en junio de 2023. [Página oficial](#).
- [34] Jose Agustín V. S. et al. Implementación de una aplicación para el análisis dosimétrico de planes de radioterapia externa basada en la api de eclipse. *Revista de Física Médica*, 22(2):37–42, 2021.
- [35] Wikipedia. definición de bolus, consultado en mayo de 2023. [Enlace](#).
- [36] Taghreed A. et al. Exaskin: A novel high-density bolus for 6mv x-rays radiotherapy. *Physica Medica*, 80:42–46, 2020.
- [37] Depositorio de github. [Enlace](#).
- [38] Wikipedia. unidades hounsfield, consultado en mayo de 2023. [Enlace](#).
- [39] Regular-expressions.info, consultado en mayo de 2023. [Enlace](#).
- [40] Foundational model of anatomy, consultado en mayo de 2023. [Enlace](#).
- [41] Álvaro B. A. y Luis S. S. Alejandro B. L., Daniel N. S. Un ejemplo de aplicación de eclipse scripting api para la automatización en la verificación de planificaciones de radioterapia. *PortalesMedicos*, 2020.
- [42] Geicam. El cáncer de mama en españa: situación actual, consultado en mayo de 2023. [Enlace](#).

## Anexo I: Código completo

### Introducción y librerías

---

```
// Por Marina Gutiérrez y Nicolás Gillard
//
// Para poder usarla, situarla en una carpeta en D:\scripts eclipse
// (crearla si no existe)
// Para ejecutarla cuando tengamos un paciente cargado vamos a
// "Scripts", en "Herramientas" y buscamos este .txt.
//
// EL USO DE ESTE SCRIPT ES SÓLO UNA AYUDA, CADA CASO MÉDICO DEBE SER
// SUPERVISADO MANUALMENTE.
////////////////////////////////////

//Primero cargamos las librerías.
using System;
using System.Linq;
using System.Text;
using System.Windows;
using System.IO;
using System.Collections;
using System.Diagnostics;
using System.Reflection;
using VMS.TPS.Common.Model.API;
using VMS.TPS.Common.Model.Types;
using System.Text.RegularExpressions;
using System.Collections.Generic;

//Vamos a crear las funciones que cubrirán cada paso.
namespace VMS.TPS
{
class Script
{
```

---

### Paso 1

---

```
static string CheckNotes(ref PlanSetup ps)
{
    return "          -Notas: " + ps.Course.Comment + "\n";
}
```

---

### Paso 3

---

```
static string CheckCT(ref ScriptContext context)
{
    //Sólo cuenta con un estudio
    if (context.Patient.Studies.Count() == 1)
```

```

{
    return "-No se ha encontrado un TC posterior.\r\n";
}
bool foundLaterStudy = false; //Se asume que no hay un estudio
    posterior
foreach(Study study in context.Patient.Studies) //Se recorre cada
    estudio
{
    if (study.CreationDateTime >
        context.PlanSetup.Series.Study.CreationDateTime) //Se
        encuentra una fecha posterior
    {
        foundLaterStudy = true; //Se ha encontrado una fecha
            posterior
        break;
    }
}
if (foundLaterStudy) //Esto es como poner if (foundLaterStudy =
    true) en Python
    return "-Se ha encontrado un TC posterior!\r\n";
else
    return "-No se ha encontrado un TC posterior.\r\n";
}

```

---

```

static string CheckName(ref StructureSet ss)
{
    string primerapellido = ss.Patient.MiddleName; //Primer apellido
    string segundoapellido = ss.Patient.LastName; //Segundo apellido
    string nombre = ss.Patient.FirstName; //Nombre
    return "-Nombre: " + primerapellido + " " + segundoapellido + ", "
        + nombre + ".\r\n";
}

```

#### Paso 4

---

```

static string CheckMarkers(ref ScriptContext context)
{
    int numberOfMarkers = 0;
    int numberOfMarkersWellPlaced = 0;
    double userX = context.Image.UserOrigin.x;
    double userY = context.Image.UserOrigin.y;
    double userZ = context.Image.UserOrigin.z;
    foreach(Structure structure in context.StructureSet.Structures)
        //Recorre las estructuras de la parte izquierda
    {
        if (structure.DicomType == "MARKER") //Se para en la que son
            de tipo "MARKER"

```

```
{
    ++numberOfMarkers; //Suma un marker al número de markers
    bool markerfound = false;
    for(int j = -20; j <=20; j+=2) //De -20 a 20 en pasos de 2
    {
        VVector start = new VVector(structure.CenterPoint.x-2,
            structure.CenterPoint.y + Convert.ToDouble(j)/10.0,
            structure.CenterPoint.z); //Se define una matriz de
            píxeles 4x4mm2 centrada en ese marcador
        VVector end = new VVector(structure.CenterPoint.x+2,
            structure.CenterPoint.y + Convert.ToDouble(j)/10.0,
            structure.CenterPoint.z);
        for(int i = 0; i < 10; i++)
        {
            if(context.Image.GetImageProfile(start, end, new
                double[10])[i].Value > 1500) //Región de alta
                densidad detectada
            {
                numberOfMarkersWellPlaced++; //Marcador bien
                colocado
                markerfound = true;
                break;
            }
        }
        if (markerfound)
        {
            break;
        }
    }
}
}
if (numberOfMarkers == 3)
{
    return "-Se han detectado los 3 marcadores que hay y están
        alineados con el origen. \r\n ";
}
else
{
    int MarkersWrongPlaced = 3 - numberOfMarkersWellPlaced;
    if ( MarkersWrongPlaced == 3 )
    {
        return "-Se han detectado los 3 marcadores que hay,
            aunque el número de marcadores alineados con el
            origen es " + numberOfMarkers + ".\r\n ";
    }
    else
    {
        return "-El número de marcadores no es el adecuado. \r\n ";
    }
}
```

```

    }
}
}

static string CheckOrigin(ref ScriptContext context)
{
    double userX=context.Image.UserOrigin.x; //Profundidad del eje X
        en el origen
    double userY=context.Image.UserOrigin.y; //Profundidad del eje Y
        en el origen
    double userZ=context.Image.UserOrigin.z; //Profundidad del eje Z
        en el origen
    if(userX==0)
    {
        return "-El origen en X, Y, Z está en (" + Math.Round(userX,2)
            + ", " + Math.Round(userY,2) + ", " + Math.Round(userZ,2) +
            "), respectivamente. Comprobar colocación del origen en
            X.\r\n";
    }
    if(userY==0)
    {
        return "-El origen en X, Y, Z está en (" + Math.Round(userX,2)
            + ", " + Math.Round(userY,2) + ", " + Math.Round(userZ,2) +
            "), respectivamente. Comprobar colocación del origen en
            Y.\r\n";
    }
    if(userZ==0)
    {
        return "-El origen en X, Y, Z está en (" + Math.Round(userX,2)
            + ", " + Math.Round(userY,2) + ", " + Math.Round(userZ,2) +
            "), respectivamente. Comprobar colocación del origen en
            Z.\r\n";
    }
    return "-El origen en X, Y, Z está en (" + Math.Round(userX,2) +
        ", " + Math.Round(userY,2) + ", " + Math.Round(userZ,2) +
        "), respectivamente.\r\n";
}
}

```

## Paso 5

```

static string CheckPrescription(ref PlanSetup ps)
{
    double dosisPorFraccionPrescripcion = 0;
    if( ps.PrimaryReferencePoint == null) //Que exista punto de
        referencia
    {
        return "-No se ha encontrado punto de referencia de
            dosis.\r\n";
    }
}

```

```
}
else if( ps.RTPrescription == null || ps.RTPrescription.Targets ==
    null) //Que tenga prescripción
{
    return "-No se ha encontrado prescripción. \r\n";
}
else
{
    //Buscamos el órgano de mayor dosis, cuya dosis, en teoría, es
    la que tiene que aparecer en el punto de referencia
    foreach (RTPrescriptionTarget t in ps.RTPrescription.Targets)
    if( t.DosePerFraction.Dose > dosisPorFraccionPrescripcion )
        //Dosis del objeto mayor que la actual
        {
            dosisPorFraccionPrescripcion = t.DosePerFraction.Dose;
            //Actualiza el valor
        }
    foreach( RTPrescriptionTarget t in ps.RTPrescription.Targets)
    if( t.DosePerFraction.Dose == dosisPorFraccionPrescripcion )
        //Verifica si el valor de la dosis por fracción en el
        objeto es igual
        {
            double suma1= t.NumberOfFractions *
                dosisPorFraccionPrescripcion; //Número de
                sesiones*dosis por sesión
            double suma2= ps.PrimaryReferencePoint.TotalDoseLimit.Dose
                ; //Número total de dosis
            double suma3 = suma2-suma1;
            if( ps.PrimaryReferencePoint.DailyDoseLimit.Dose !=
                dosisPorFraccionPrescripcion ||
                ps.PrimaryReferencePoint.SessionDoseLimit.Dose !=
                dosisPorFraccionPrescripcion || suma3>0.0001) //El
                número de sesiones*dosis por sesión no coincide con la
                dosis total.
            {
                return "-Los limites de dosis no coinciden con la
                    prescripción. La dosis por fracción en el punto de
                    referencia de dosis es " +
                    ps.PrimaryReferencePoint.DailyDoseLimit.Dose + "
                    cGy mientras que en la prescripción es " +
                    dosisPorFraccionPrescripcion + " cGy.\r\n";
            }
        }
    else
    {
        return "-Los limites de dosis coinciden con la
            prescripción, en el punto de referencia la dosis
            por fracción es " +
            ps.PrimaryReferencePoint.DailyDoseLimit.Dose + "
```

```

        cGy y en la prescripción es " +
        dosisPorFraccionPrescripcion + " cGy. La dosis
        total es " +
        t.NumberOfFractions*t.DosePerFraction.Dose + " cGy,
        dividido en " + t.NumberOfFractions + "
        sesiones.\r\n";
    }
}
}
return "-La prescripción es válida, aunque es MUY CONVENIENTE
comprobarla. \r\n"; //Si la función ha llegado hasta aquí, no
debería haber problema
}

```

## Paso 6

```

static string CheckCouch(ref StructureSet ss , ref PlanSetup ps)
{
    bool couchInterior = true; //Los booleanos se encargan de
        comprobar si existen o no
    bool couchSurface = true;
    int interior = 0; //Los enteros se encargan de comprobar si están
        duplicadas o no
    int surface = 0;
    foreach(Structure structure in ss.Structures) //Recorre cada
        estructura
    {
        if (structure.Id == "CouchInterior") //Se llama CouchInterior
            couchInterior = false;
        if (structure.Id == "CouchSurface")
            couchSurface = false;
        if (structure.Id.Contains("CouchInterior")) //A ade un punto
            por cada estructura que contenga CouchInterior, idealmente
            sólo debería haber una, en cuyo caso el contador
            permanecería en 1.
            interior++;
        if (structure.Id.Contains("CouchSurface"))
            surface++;
    }
    if( couchInterior ) //Eso significa que couchInterior = true, que
        sólo sucede si no se encuentra la estructura llamada
        CouchInterior
        return "-No hay couchInterior, es cabeza?\r\n";
    if( couchSurface )
        return "-No hay couchSurface, es cabeza?\r\n";
    if(interior > 1) //Se han encontrado varias estructuras que
        contengan "Couch interior"
        return "-CouchInterior está duplicado.\r\n";
}

```

```

    if(surface > 1)
        return "-CouchSurface está duplicado.\r\n";
    else
        return "-Se ha comprobado que existen las estructuras
                CouchInterior y CouchSurface.\r\n"; //Y no están duplicadas.
}
}

static string CheckMaquinaMesa(ref StructureSet ss , ref PlanSetup ps)
{
    string maquinaid = "";
    string maquinamodel = "";
    foreach(Beam beam in ps.Beams)
    {
        maquinaid = beam.TreatmentUnit.Id; //Id de la máquina
        maquinamodel = beam.TreatmentUnit.MachineModel; //Modelo de la
            máquina
    }
    foreach(Structure structure in ss.Structures) //Se recorren las
        estructuras
    {
        if ( structure.Id == "CouchInterior" ) //Sólo si existe la
            estructura CouchInterior
        {
            return "-El id de la máquina es " + maquinaid + ", modelo
                " + maquinamodel + ".\r\n";
        }
    }
    return "-No se puede proporcionar el id de la máquina porque no se
        ha encontrado ninguna estructura con el identificador
        \"CouchInterior\".";
}
}

```

## Paso 7

```

static string CheckVMATorIMRT(ref PlanSetup ps)
{
    string tecnica = "";
    foreach (Beam beam in ps.Beams)
    {
        tecnica = beam.MLCPlanType.ToString(); //Lee la técnica
            empleada en cada campo (imrt o VMAT)
        if (tecnica == "imrt" && beam.Id.Contains("Campo")) //Si es
            imrt y el nombre del campo contiene "Campo"
            return "-Hay un error, la técnica empleada es imrt pero el
                Id del campo, en vez de ser el ángulo, contiene la
                palabra Campo.\r\n";
        if (tecnica == "imrt") //Si ha llegado hasta aquí, es que es
            imrt y no se llama "Campo".
    }
}

```

```

        return "-La técnica empleada es imrt, lo cual es correcto
                porque el campo tiene el nombre del ángulo. \r\n";
    if(tecnic == "VMAT" && beam.Id.Contains("Campo")) //Si es
        VMAT y contiene "Campo" o "Arco"
        return "-La técnica de tratamiento está siendo VMAT, lo
                cual es correcto porque el campo se llama Campo +
                número. \r\n";
    if(tecnic == "VMAT" && beam.Id.Contains("Arco"))
        return "-La técnica de tratamiento está siendo VMAT, lo
                cual es correcto porque el campo se llama Arco +
                número. \r\n";
    else
        return "Hay algo mal, la técnica de tratamiento es " +
                beam.MLCPlanType + " y el ID del campo es " + beam.Id +
                ". Con técnica VMAT el nombre debe ser Campo 1, 2... y
                con IMRT el nombre debe corresponderse con el ángulo.
                La técnica o el nombre no son correctos. \r\n";
    }
    return "No se ha detectado ni técnica imrt ni VMAT, se trata de una
            diferente? De no ser así, REVISAR.\r\n";
}

```

## Paso 9

```

static string CheckMismoAcelerador(ref PlanSetup ps)
{
    string primerAcelerador = "";
    bool aceleradoresIguales = true; //Partimos de que todos los
        aceleradores son iguales
    foreach(Beam beam in ps.Beams) //Recorremos cada campo
    {
        if(string.IsNullOrEmpty(primerAcelerador))
        {
            primerAcelerador = beam.TreatmentUnit.Id;
        }
        else if(!beam.TreatmentUnit.Id.Equals(primerAcelerador)) //El
            acelerador de este paso no ha coincidido con el primero
        {
            aceleradoresIguales = false; //Los aceleradores no
                coinciden
            break; //Se fuerza el fin del bucle
        }
    }
    if (aceleradoresIguales) //Bool = true, en ningún momento se ha
        dado la condición de que no sean iguales
    {
        return "-Todos los aceleradores presentes son " +
                primerAcelerador + " y coinciden. \r\n";
    }
}

```

```
}  
return "-LOS ACELERADORES NO COINCIDEN. \r\n"; //Si se ha llegado  
aquí es que no se ha cumplido el if anterior, en cuyo caso no  
coinciden.  
}
```

---

```
static string CheckMismoIsocentro(ref PlanSetup ps)  
{  
    string primerIsocentroX = "";  
    bool isocentrosIgualesX = true;  
    double xfinal = 0.0;  
    double yfinal = 0.0;  
    double zfinal = 0.0;  
    foreach(Beam beam in ps.Beams)  
    {  
        if(string.IsNullOrEmpty(primerIsocentroX))  
        {  
            double x = beam.IsocenterPosition.x;  
            string primerisocentroX = x.ToString();  
        }  
        else if (!beam.IsocenterPosition.x.Equals(primerIsocentroX))  
            //El valor X de los isocentros no coinciden por lo que el  
            bool de X es ahora false.  
        {  
            isocentrosIgualesX = false;  
            break;  
        }  
    }  
    string primerIsocentroY = "";  
    bool isocentrosIgualesY = true;  
    foreach(Beam beam in ps.Beams)  
    {  
        if(string.IsNullOrEmpty(primerIsocentroY))  
        {  
            double y = beam.IsocenterPosition.y;  
            string primerisocentroY = y.ToString();  
        }  
        else if (!beam.IsocenterPosition.y.Equals(primerIsocentroY))  
            //El valor Y de los isocentros no coinciden por lo que el  
            bool de Y es ahora false.  
        {  
            isocentrosIgualesY = false;  
            break;  
        }  
    }  
    string primerIsocentroZ = "";  
    bool isocentrosIgualesZ = true;  
    foreach(Beam beam in ps.Beams)
```

```
{
    if(string.IsNullOrEmpty(primerIsocentroZ))
    {
        double z = beam.IsocenterPosition.z;
        string primerisocentroZ = z.ToString();
    }
    else if (!beam.IsocenterPosition.z.Equals(primerIsocentroZ))
        //El valor Z de los isocentros no coinciden por lo que el
        //bool de Z es ahora false.
    {
        isocentrosIgualesZ = false;
        break;
    }
}
if (isocentrosIgualesX && isocentrosIgualesY &&
isocentrosIgualesZ) //Los tres bool = true, todas las
componentes de cada campo coinciden.
{
    foreach(Beam beam in ps.Beams)
    {
        xfinal = beam.IsocenterPosition.x;
        yfinal = beam.IsocenterPosition.y;
        zfinal = beam.IsocenterPosition.z;
        if (xfinal.ToString().EndsWith("0") &&
            yfinal.ToString().EndsWith("0") &&
            zfinal.ToString().EndsWith("0")) //Que sean decimales
            perfectos
        {
            return "-Todos los isocentros coinciden y son
                decimales perfectos. El valor del isocentro es (" +
                Math.Round(xfinal,2) + ", " + Math.Round(yfinal,2)
                + ", " + Math.Round(zfinal,2) + ")\n";
        }
        else
        {
            return "-Los isocentros coinciden, pero no son
                decimales perfectos.\n";
        }
    }
}
else
{
    return "-LOS ISOCENTROS NO COINCIDEN \r\n";
}
return "-Algo ha salido mal. \r\n";
}
```

## Paso 11

---

```
static string CheckCalculationModel(ref PlanSetup ps)
{
    if(ps.PhotonCalculationModel.Contains("Acurous"))
        return "-El modelo utilizado es " + ps.PhotonCalculationModel
            + ", que puede ser el adecuado. \r\n";
    if(ps.PhotonCalculationModel.Contains("Acuros"))
        return "-El modelo utilizado es " + ps.PhotonCalculationModel
            + ", que puede ser el adecuado. \r\n";
    if(ps.PhotonCalculationModel.Contains("AAA"))
        return "-El modelo utilizado es " + ps.PhotonCalculationModel
            + ", que puede ser el adecuado. \r\n";
    else
        return "-El modelo de cálculo es " + ps.PhotonCalculationModel
            + ", en principio, los adecuados son AcurosXB o AAA,
            comprobar esto.\r\n";
}
```

---

## Paso 12

---

```
static string CheckColisiones(ref PlanSetup ps)
{
    string info1 = ""; //Dos posibles mensajes
    string info2 = "";
    foreach(Beam beam in ps.Beams) //Recorre cada campo
    {
        double rotacion =
            beam.ControlPoints.First().PatientSupportAngle; //en cada
            campo, lee el ángulo de CouchRTN
        if(rotacion!=0 && rotacion<=3) //Si CouchRTN es superior a
            cero e inferior a tres grados en algún campo
        {
            info1 = "El Couch rotation (CouchRTN) es " + rotacion + ",
                que es distinto de 0, aunque la probabilidad es baja,
                podría llegar a haber colisión.\n";
        }
        if(rotacion >= 3) //Si CouchRTN es superior a tres grados en
            algún campo
        {
            info2 = "El Couch rotation (CouchRTN) es " + rotacion + ",
                que es superior a 3, podría llegar a haber colisión.\n";
        }
    }
    if (String.IsNullOrEmpty(info1) && String.IsNullOrEmpty(info2))
        //En ningún campo se ha detectado una rotación superior a cero
        grados
        return "-El Couch Rotation (CouchRTN) es 0 en todos los
            campos, hay una probabilidad de colisión ínfima.\n";
}
```

```

else if(String.IsNullOrEmpty(info2)) //En ningún campo se ha
    detectado una rotación superior a tres grados
    return info1;
else //En algún campo se ha detectado rotación superior a tres
    grados
    return info2;
}

```

### Paso 13 (sin indentados para que no ocupe demasiado)

```

static string CheckConstraints(ref PlanSetup ps, ref StructureSet ss,
    ref Image image, ref RTPrescription rtprescription){
if(rtprescription == null || !ps.IsDoseValid ) // Realmente
    AutochecklistCore ya se llama con una dosis válida
return "- Imposible comprobar los constraints, debido a problemas con
    la prescripción o la dosis.\r\n";

//Creamos el diccionario con todos los términos que se nos ocurra
var dict = new Dictionary<string, List<string>>();
dict.Add("7647", new List<string>(new string[] {@"^\s*MED.*",
    @"^\s*SPINAL.*CORD.*"}));
dict.Add("7131", new List<string>(new string[] {@"^\s*ESF.*",
    @"^\s*ESO.*"}));
dict.Add("7088", new List<string>(new string[] {@"^\s*COR.*",
    @"^\s*HEART.*"}));
dict.Add("14544", new List<string>(new string[] {@"^\s*REC.*"}));
dict.Add("16950", new List<string>(new string[] {@"SNF.*", @"SINF.*",
    @"SYMPH.*"}));
dict.Add("15900", new List<string>(new string[] {@"^\s*VEJ.*",
    @"^\s*BLA.*", @"^\s*BLD.*"}));
dict.Add("7199", new List<string>(new string[]
    {@"^\s*IND.*",@"^\s*INT.*", @"BOW.*", @"DELGADO"}));
dict.Add("7200", new List<string>(new string[]
    {@"^\s*IND.*",@"^\s*INT.*", @"BOW.*", @"DELGADO"}));
dict.Add("7197", new List<string>(new string[] {@"^\s*HID.*",
    @"^\s*HIG.*", @"^\s*LIV.*"}));
dict.Add("62045", new List<string>(new string[] {@"^\s*QUI.*",
    @"^\s*CHIAS.*"}));
dict.Add("79876", new List<string>(new string[] {@"^\s*TRC.*",
    @"^\s*TRONC.*", @"^\s*BRAIN.*STEM.*"}));
dict.Add("55097", new List<string>(new string[] {@"^\s*LAR.*"}));
dict.Add("9603", new List<string>(new string[] {@"^\s*TIR.*",
    @"^\s*THYR.*"}));
dict.Add("52748", new List<string>(new string[] {@"^\s*MAN.*",
    @"^\s*MND.*"}));
dict.Add("46688", new List<string>(new string[] {@"^\s*FAR.*",
    @"^\s*CONSTRICT.*FAR.*", @"^\s*PHAR.*"}));
dict.Add("7394", new List<string>(new string[] {@"^\s*TR[AÁ].*"}));

```

```

dict.Add("20292", new List<string>(new string []
    {@"^\s*BOC.*",@"^\s*CAV.*BUC.*", @"^\s*CAV.*OR.*",
    @"^\s*OR.*CAV.*"}));
dict.Add("62000", new List<string>(new string [] {@"^\s*CER.*",
    @"^\s*BRAIN\s*"}));
dict.Add("19614", new List<string>(new string [] {@"^\s*BUL.*",
    @".*PENEAN.*", @".*PENIL.*"}));
dict.Add("3862", new List<string>(new string [] {@"^\s*LAD"}));
dict.Add("7405", new List<string>(new string [] {@"^.*BRON.*PR.*",
    @"^.*MAIN.*BR\.*"}));
dict.Add("7196", new List<string>(new string [] {@"^.*SPLEEN.*",
    @"^\s*BAZ\.*"}));
dict.Add("13889", new List<string>(new string [] {@"^.*HIP[OÓ]FIS.*",
    @"^\s*PITUI\.*"}));
dict.Add("7148", new List<string>(new string [] {@"^.*EST[OÓ].*",
    @".*STOM\.*"}));
dict.Add("59815", new List<string>(new string [] {@"^\s*LAB.*",
    @"^\s*LIP.*"}));
dict.Add("275020",new List<string>(new string [] {@"^\s*HIP.*CAMP.*"}));
dict.Add("GreatVessels",new List<string>(new string [] {@"^.*Vasos.*",
    @"^.*Vessels.*"}));
dict.Add("19667",new List<string>(new string [] {@"^.*Uret.*"}));
dict.Add("71331", new List<string>(new string [] { @"^.*Cost.*"}));
dict.Add("5906",new List<string>(new string [] {@"PLEX"}));
dict.Add("60202",new List<string>(new string []
    {@"C[OÓ]C(H)?L.*[~I][DR].*", @"C[OÓ]C(H)?LEA\s*$"}));
dict.Add("60203",new List<string>(new string []
    {@"C[OÓ]C(H)?L.*[IL].*", @"C[OÓ]C(H)?LEA\s*$"}));
dict.Add("19908", new List<string>(new string []
    {@"^\s*MAM.*[~I][DR].*", @"^\s*BREAST.*[~I][DR].*"}));
dict.Add("19910", new List<string>(new string [] {@"^\s*MAM.*[IL].*",
    @"^\s*BREAST.*[IL].*"}));
dict.Add("55011", new List<string>(new string []
    {@"^\s*CF.*[~I][DR].*",@"^\s*CABEZA FEMORAL DCH.*",
    @"^\s*CFM.*[~I][DR].*", @"CABEZAS\s*FEMORALES.*",
    @"FEMORAL\s*HEAD.*"}));
dict.Add("55012", new List<string>(new string []
    {@"^\s*CF[IL].*",@"^\s*CABEZA FEMORAL IZQ.*",@"^\s*CFM.*[IL].*",
    @"CABEZAS\s*FEMORALES.*", @"FEMORAL\s*HEAD.*"}));
dict.Add("7309", new List<string>(new string [] {@"^.*PUL.*[~I][DR].*",
    @".*LUNG.*[~I][DR].*"}));
dict.Add("7310", new List<string>(new string [] {@"^.*PUL.*[IL].*",
    @".*LUNG.*[IL].*"}));
dict.Add("68877", new List<string>(new string [] {@"^.*PULMONES.*",
    @".*LUNGS.*", @".*PUL$", @".*PULMON$", @".*LUNG$"}));
dict.Add("12515", new List<string>(new string []
    {@"^\s*EYE\s*[_-]*\s*[IL].*", @"^\s*OJO.*[IL].*",
    @"^\s*GLOBO.*[IL].*"}));

```

```

dict.Add("12514", new List<string>(new string[]
    {@"^\s*EYE\s*[_-]*\s*[\^I][DR].*", @"^\s*OJO.*[\^I][DR].*",
    @"^\s*GLOBO.*[\^I][DR].*"}));
dict.Add("58243", new List<string>(new string[]
    {@"^\s*CRI\s*[_-]*\s*[IL].*", @"^\s*LEN.*[IL].*",
    @"^\s*CRISTALIN.*[IL].*"}));
dict.Add("58242", new List<string>(new string[]
    {@"^\s*CRI\s*[_-]*\s*[\^I][DR].*", @"^\s*LEN.*[\^I][DR].*",
    @"^\s*CRISTALIN.*[\^I][DR].*"}));
dict.Add("50878", new List<string>(new string[]
    {@"^\s*VOP\s*[_-]*\s*[IL].*",
    @"^\s*N\s*\.*\s*[OÓ]PTICO\s*[_-]*\s*[IL].*",
    @"^\s*OPT.*NERVE.*[IL].*", @"^\s*VOP\s*$",
    @"^\s*N\s*\.*\s*[OÓ]PTICO\s*$", @"^\s*OPT.*NERVE\s*$"}));
dict.Add("50875", new List<string>(new string[]
    {@"^\s*VOP\s*[_-]*\s*[\^I][DR].*",
    @"^\s*N\s*\.*\s*[OÓ]PTICO\s*[_-]*\s*[\^I][DR].*",
    @"^\s*OPT.*NERVE.*[\^I][DR].*", @"^\s*VOP\s*$",
    @"^\s*N\s*\.*\s*[OÓ]PTICO\s*$", @"^\s*OPT.*NERVE\s*$"}));
dict.Add("59798", new List<string>(new string[]
    {@"^\s*PAR\s*[_-]*\s*[IL].*",@"PAR[OÓ]TID.*[IL].*",
    @"PAR[OÓ]TID[A]*S.*"}));
dict.Add("59797", new List<string>(new string[]
    {@"^\s*PAR\s*[_-]*\s*[DR].*",@"PAR[OÓ]TID.*[DR].*",
    @"PAR[OÓ]TID[A]*S.*"}));
dict.Add("7204", new List<string>(new string[]
    {@"^\s*R[EI][N ][_-]D.*", @"^\s*RI[N ][OÓ]N D.*"}));
dict.Add("7205", new List<string>(new string[]
    {@"^\s*R[EI][N ][_-]I.*", @"^\s*RI[N ][OÓ]N I.*"}));
dict.Add("59803", new List<string>(new string[]
    {@"^\s*GSM[_-]I.*",@"^\s*GL[AÁ]NDULA.*SU.*"}));
dict.Add("59802", new List<string>(new string[]
    {@"^\s*GSM[_-]D.*",@"^\s*GL[AÁ]NDULA.*SU.*"}));
dict.Add("7101", new List<string>(new string[]
    {@"^\s*Ventr.*[IL].*"}));

string lines = "\r\n--- LIMITES DE DOSIS ---\r\n\n"; //Se crean los
    textos a devolver, por ahora vacíos
string lines2 = "";
string lines3 = "";
string tipoConstraint = "";
bool limitesSinCumplir = false;

// Convertimos contra/ipsi a izquierda/derecha
string ipsilateral = "";
string contralateral = "";
foreach(Structure structure in ss.Structures)
if(structure.DicomType == "PTV")

```

```
if (structure.CenterPoint.x - image.UserOrigin.x > 0){
    ipsilateral = "I";
    contralateral = "D";}
else{
    ipsilateral = "D";
    contralateral = "I";}

foreach(RTPrescriptionOrganAtRisk oar in rtprescription.OrgansAtRisk){
    // Identificamos el OAR de la prescripción
    List<string> OARkeys = new List<string>();
    foreach (KeyValuePair<string, List<string>> dictEntry in dict){
        foreach(string str in dictEntry.Value){
            string organAtRiskId = oar.OrganAtRiskId;
            organAtRiskId = Regex.Replace(organAtRiskId, @"IPSI.*", ipsilateral,
                RegexOptions.IgnoreCase);
            organAtRiskId = Regex.Replace(organAtRiskId, @"CONTRAL.*",
                contralateral, RegexOptions.IgnoreCase);
            Regex regex = new Regex(str, RegexOptions.IgnoreCase);
            if( regex.IsMatch(organAtRiskId) )
                OARkeys.Add(dictEntry.Key);}}

    // Comprobamos que lo haya encontrado
    if(!OARkeys.Any()){
        lines2 += " * No se entiende a qué estructura se refiere la
            prescripción " + oar.OrganAtRiskId + "\r\n";

        continue;}

    // Identificamos el OAR (Organs at risk) en las estructuras
    List<string> OARIds = new List<string>();
    foreach(Structure structure in ss.Structures){
        foreach(StructureCodeInfo structureCodeInfo in
            structure.StructureCodeInfos)
            foreach(string OARkey in OARkeys)
                if(structureCodeInfo.Code == OARkey)
                    OARIds.Add(structure.Id);}

    // Comprobamos que lo haya encontrado
    if(!OARIds.Any()){
        lines2 += " * No se ha encontrado la estructura correspondiente a la
            prescripción " + oar.OrganAtRiskId + "\r\n";
        continue;}

    // Comprobamos los constraints
    foreach(Structure structure in ss.Structures){
        foreach(string OARId in OARIds){
            // Evitamos las estructuras creadas por física
            if(structure.Id != OARId || structure.Id.StartsWith(@"F_") ||
```

```

    structure.Id.StartsWith(@"NS_"))
continue;

foreach(RTPrescriptionConstraint constraint in oar.Constraints){
Regex space = new Regex(@"^\s+$");

// Constraint tipo texto libre
if( constraint.ConstraintType == RTPrescriptionConstraintType.FreeText
    && !space.IsMatch(constraint.Value2) && constraint.Value1.Any())
{
string constVal1 = constraint.Value1;
constVal1.Replace(",", ".");
Regex rgxV = new Regex(@"V\s*(\d*\.\d*)\s*"); // Vx.x...
Regex rgxVp = new Regex(@"V\s*(\d*\.\d*)\s*\%"); // Vx.x%...
Regex rgxD = new Regex(@"D\s*(\d*\.\d*)\s*"); // Dx.x...

Regex rgxDcc = new Regex(@"D\s*(\d*\.\d*)\s*cc"); // Dx.xcc

Regex rgxDp = new Regex(@"D\s*(\d*\.\d*)\s*\%"); // Dx.x%
Regex rgxP = new Regex(@"^\s*(\d*\.\d*)\s*\%"); // x.x%
Regex rgxCC = new Regex(@"^\s*(\d*\.\d*)\s*cc"); // x.xcc
Regex rgx = new Regex(@"^\s*(\d*\.\d*)\s*"); // x.x...
Regex rg = new Regex(@"(\d*\.\d+)"); // ...x.x...
double aux = 100000;

try{
if( !rgxD.IsMatch(constVal1) && !rgxCC.IsMatch(constVal1) &&
    !rgxCC.IsMatch(constraint.Value2) ) // VGy = % include Vx.x...
    = x.x % Vx.x... = x.x x.x = x.x % x.x = x.x
aux = ps.GetVolumeAtDose( structure, new
    DoseValue(Convert.ToDouble(rg.Match(constVal1).Groups[1].Value), "cGy")*100,
    VolumePresentation.Relative);

else if( !rgxD.IsMatch(constVal1) && !rgxCC.IsMatch(constVal1) &&
    rgxCC.IsMatch(constraint.Value2) ) // VGy = cc include Vx.x...
    = x.x cc x.x = x.x cc
aux = ps.GetVolumeAtDose( structure, new
    DoseValue(Convert.ToDouble(rg.Match(constVal1).Groups[1].Value), "cGy")*100,
    VolumePresentation.AbsoluteCm3);
else if( rgxVp.IsMatch(constVal1) && !rgxCC.IsMatch(constraint.Value2)
    ) // V% = % include Vx.x % = x.x Vx.x % = x.x %
aux = ps.GetVolumeAtDose( structure, new
    DoseValue(Convert.ToDouble(rg.Match(constVal1).Groups[1].Value), "Percent"),
    VolumePresentation.Relative);

else if( rgxVp.IsMatch(constVal1) && rgxCC.IsMatch(constraint.Value2)
    ) // V% = cc include Vx.x % = x.x cc

```

```

aux = ps.GetVolumeAtDose(          structure, new
    DoseValue(Convert.ToDouble(rg.Match(constVal1).Groups[1].Value), "Percent"),
    VolumePresentation.AbsoluteCm3);
else if( rgxD.IsMatch(constVal1) && !rgxDcc.IsMatch(constVal1) ||
    rgxP.IsMatch(constVal1) ) // D% = Gy    incluye    Dx.x % = ...
    Dx.x = ...    x.x % = ...
aux = ps.GetDoseAtVolume(          structure,
    Convert.ToDouble(rg.Match(constVal1).Groups[1].Value),
    VolumePresentation.Relative,
    DoseValuePresentation.Absolute).Dose;

else if( rgxDcc.IsMatch(constVal1) || rgxCC.IsMatch(constVal1)
    ) // Dcc = Gy    incluye    Dx.x cc = ...    x.x cc =
    ...
    {
aux = ps.GetDoseAtVolume(          structure,
    Convert.ToDouble(rg.Match(constVal1).Groups[1].Value),
    VolumePresentation.AbsoluteCm3,
    DoseValuePresentation.Absolute).Dose;}
string atencion = "";
if(aux >
    Convert.ToDouble(rg.Match(constVal1).Groups[1].Value)){

limitesSinCumplir = true;
atencion = "ATENCIÓN: ";}
lines += " * " + atencion + "Prescrito: " + oar.OrganAtRiskId
+ " " + constVal1 + " " + constraint.Unit1 + " < " + constraint.Value2
+ " " + constraint.Unit2 + "\t";
lines += "Conseguido: " + structure.Id + " " + aux.ToString("0.##") +
    "\r\n";}
catch{
lines3 += " * Error con la prescripción de " + oar.OrganAtRiskId +
    ": " + oar.OrganAtRiskId
+ " " + constVal1 + " " + constraint.Unit1 + " < " + constraint.Value2
+ " " + constraint.Unit2 + "\r\n";}

// Constraint tipo predefinido
else if( constraint.ConstraintType !=
    RTPrescriptionConstraintType.FreeText &&
    !space.IsMatch(constVal1) && constraint.Value1.Any()){

try{
double aux = 100000;
if(constVal1.ConstraintType ==
    RTPrescriptionConstraintType.MaximumMeanDose)
aux = ps.GetDVHCumulativeData(structure,
    DoseValuePresentation.Absolute, VolumePresentation.Relative, 0.001
    ).MeanDose.Dose;
if(constVal1.ConstraintType ==

```

```

    RTPrescriptionConstraintType.MaximumDose)
aux = ps.GetDVHCumulativeData(structure,
    DoseValuePresentation.Absolute, VolumePresentation.Relative, 0.001
    ).MaxDose.Dose;
if(constraint.ConstraintType ==
    RTPrescriptionConstraintType.MaximumMeanDose)
tipoConstraint = "<D>";
if(constraint.ConstraintType ==
    RTPrescriptionConstraintType.MaximumDose)
tipoConstraint = "Dmax";
aux = aux/100 //Pasamos a cGy
string atencion = "";
if(aux > Convert.ToDouble(constraint.Value1)){
limitesSinCumplir = true;
atencion = "ATENCIÓN: ";}
lines += " * " + atencion + "Prescrito: " + oar.OrganAtRiskId
+ " " + tipoConstraint + " < " + constraint.Value1 + " " +
    constraint.Unit1 + "\t";
lines += "Conseguido: " + structure.Id + " " + aux.ToString("0.##") +
    "\r\n";
}

catch{
lines3 += " * Error con la prescripción de " + oar.OrganAtRiskId +
    ": " + oar.OrganAtRiskId
+ " " + tipoConstraint + " < " + constraint.Value1 + " " +
    constraint.Unit1 + "\r\n";}}}}}}

if(limitesSinCumplir)
lines = "Existen OAR que no cumplen los límites de dosis.\r\n" + lines;
return lines + "\n" + lines2 + "\n" + lines3;}

```

---

## Paso 15

```

static string CheckUMcGy (ref PlanSetup ps)
{
    foreach(Beam beam in ps.Beams) //Se recorre cada campo
    {
        if(beam.Meterset.Value >= 1500) //Valor de UM superior a 1500
            return "-El valor máximo de UM en la máquina es " +
                beam.Meterset.Value + ", lo cual SOBREPASA los límites
                (1500), es necesario dividirlo.\n";
    }
    return "-En ninguno de los campos se sobrepasa el valor de
        Unidades Monitor (UM), cuyo límite es 1500.\n";
}

```

---

## Paso 16

---

```

static string CheckBolus (ref PlanSetup ps)
{
    string primerBolus = "";
    bool bolusIguales = true; //Se parte de que todos los bolus son
        iguales
    string thickness = "";
    foreach(Beam beam in ps.Beams) //Se recorre cada campo
    {
        if(string.IsNullOrEmpty(primerBolus))
        {
            primerBolus = ps.RTPrescription.BolusThickness; //Si es el
                primer bolus, se toma su valor
        }
        else if(!ps.RTPrescription.BolusThickness.Equals(primerBolus))
            //Si el valor no coincide con el del primer bolus
        {
            bolusIguales = false; //Los bolus no han coincidido
            break;
        }
    }
    if (bolusIguales) //Ya sea porque no hay (coinciden al ser 0) o
        porque hay y coinciden
    {
        foreach(Beam beam in ps.Beams)
        {
            thickness = ps.RTPrescription.BolusThickness;
            if (String.IsNullOrEmpty(thickness)) //Si el espesor es 0
                es que han coincidido por ser ambos 0
                return "-No hay boluses.\n";
            else //Han coincidido y sus espesores son distintos de 0
                return "-Se han encontrado boluses, es el mismo en
                    todos los aceleradores y su espesor es " +
                    thickness + ".\n";
        }
    }
    return "-LOS BOLUSES NO COINCIDEN. \r\n"; //A esto sólo se llega
        si no se da la condición de bolusIguales = true
}

```

---

### Paso (adicional) 17

---

```

static string CheckStructures(ref StructureSet ss)
{
    foreach(Structure structure in ss.Structures) //Recorre cada
        estructura
        if( structure.IsEmpty ) //Si una estructura está vacía
        {

```

```

        return "-La estructura " + structure.Id + " está vacía.
            \r\n";
    }
    bool existeBody = false; //Se asume que no existe BODY
    foreach(Structure structure in ss.Structures)
    {
        if(structure.Id == "BODY") //Se encuentra BODY
            existeBody = true; //Existe BODY
        if( structure.Id == "BODY" && structure.DicomType !=
            "EXTERNAL") //Existe pero no es del tipo EXTERNAL
            return "-Existe la estructura " + structure.Id + ", pero
                no es de tipo EXTERNAL, es de tipo " +
                structure.DicomType + ". \r\n";
    }
    if(!existeBody) //El bool = false porque no se ha dado la
        condición que hace que sea true
    {
        return "-No existe estructura llamada BODY. Comprobar esto.
            \r\n";
    }
    return "-Bien, no hay ninguna estructura vacía, existe la
        estructura BODY y es de tipo EXTERNAL.\n";
}

```

## Returns reducidos

```

////////////////////////////////FINAL////////////////////////////////
//Se comprueba primero que el contexto sea correcto, es decir, existe
    tanto el paciente como el plan.
public void Execute(ScriptContext context)
{
    if (context.Patient == null || context.Course == null ||
        context.PlanSetup == null)
    {
        MessageBox.Show("No se ha cargado ningún plan.");
        return;
    }

    //Se definen las variables ss y ps, que contienen el StructureSet
        y PlanSetup.
    StructureSet ss = context.StructureSet;
    PlanSetup ps = context.PlanSetup;
    Image image = ss.Image;
    RTPrescription rtPrescription = ps.RTPrescription;

    //3 Correspondencia del nombre del paciente, prescripción y etapa
        tratamiento

```

```
string etapa = CheckCT(ref context);
string etapafinal = "";
if(etapa.Contains("Se ha encontrado un TC posterior"))
    etapafinal = "          -INCORRECTA etapa\n";
else
{
    etapafinal = "          -Correcta etapa\n";
}
string nombrepaciente = CheckName(ref ss);

//4- Verificación origen PTs
string markers = CheckMarkers(ref context);
string markersfinal = "";
if(markers.Contains("El número de marcadores no es el adecuado."))
    markersfinal = "          -INCORRECTO marcadores\n";
else
{
    markersfinal = "          -Correcto marcadores\n";
}
string origin = CheckOrigin(ref context);

//5- Verificación de dosis de prescripción y número de sesiones
correcto
string prescription = CheckPrescription(ref ps);
string prescriptionfinal = "";
if(prescription.Contains("Los limites de dosis no coinciden"))
    prescriptionfinal = "          -INCORRECTA prescripción\n";

else
{
    prescriptionfinal = "          -Correcta prescripción\n";
}

//6- Correcta posición y mesa de tratamiento
string position = CheckCouch(ref ss , ref ps);
string positionfinal = "";
if(position.Contains("No hay") || position.Contains("está
duplicado"))
    positionfinal = "          -INCORRECTA mesa (a menos que sea
cabeza)\n";
else
{
    positionfinal = "          -Correcta mesa\n";
}

string mesa = CheckMaquinaMesa(ref ss , ref ps);

//7- Técnica de tratamiento adecuada según prescripción
```

```
string tecnica = CheckVMATorIMRT(ref ps);
string tecnicafinal = "";
if(tecnica.Contains("Hay algo mal") || tecnica.Contains("Hay un
    error") || tecnica.Contains("No se ha detectado"))
    tecnicafinal = "                -INCORRECTA técnica\n";
else
{
    tecnicafinal = "                -Correcta técnica\n";
}

//8- Posición correcta del isocentro
string isocentro = CheckMismoIsocentro(ref ps);
string isocentrofina = "";
if(isocentro.Contains("-LOS ISOCENTROS NO COINCIDEN") ||
    isocentro.Contains("Algo ha salido mal "))
    isocentrofina = "                -INCORRECTO isocentro\n";
else
{
    isocentrofina = "                -Correcto isocentro\n";
}

//9- Todos los campos asociados al mismo acelerador y con el mismo
    isocentro
string acelerador = CheckMismoAcelerador(ref ps);
string aceleradorfinal = "";
if(acelerador.Contains("-LOS ACELERADORES NO COINCIDEN"))
    aceleradorfinal = "                -INCORRECTO acelerador\n";

else
{
    aceleradorfinal = "                -Correcto acelerador\n";
}

//11- Verificación algoritmo de cálculo
string algoritmo = CheckCalculationModel(ref ps);
string algoritmofinal = "";
if(algoritmo.Contains("en principio"))
    algoritmofinal = "                -INCORRECTO algoritmo(?)\n";

else
{
    algoritmofinal = "                -Correcto algoritmo\n";
}

//12- Verificación que no hay colisiones con mesa, soportes y/o
    paciente
string choque = CheckColisiones(ref ps);
string choquefinal = "";
```

```
if(choque.Contains("podría llegar a haber"))
    choquefinal = "          -PROBABILIDAD de colisión (?)\n";

else
{
    choquefinal = "          -Improbable colisión\n";
}

//13) Verificar que se cumplen los objetivos y restricciones
string limites = CheckConstraints(ref ps, ref ss, ref image, ref
    rtPrescription);

//15- Verificar que no se exceden los límites de UM de la máquina
string UM = CheckUMcGy(ref ps);
string UMfinal = "";
if(UM.Contains("El valor máximo de UM"))
    UMfinal = "          -INCORRECTAS UM\n";
else
{
    UMfinal = "          -Correctas UM\n";
}

//16- Bolus
string bolus = CheckBolus(ref ps);
string bolusfinal = "";
if(bolus.Contains("LOS BOLUSES NO COINCIDEN"))
    bolusfinal = "          -INCORRECTO bolus\n";
else
{
    bolusfinal = "          -Correcto bolus\n";
}

//17- Comprobación de que no hay estructuras vacías y existe una
    estructura body de tipo EXTERNAL
string structures = CheckStructures(ref ss);
string structuresfinal = "";
if(structures.Contains("pero no es de tipo EXTERNAL") ||
    structures.Contains("está vacía") || structures.Contains("No
    existe estructura llamada BODY"))
    structuresfinal = "          -INCORRECTAS estructuras\n";

else
{
    structuresfinal = "          -Correctas estructuras\n";
}
```

---

## Prints finales

---

```
//Primera ventaja emergente (todos los pasos menos el 13)
string PrintFinal1 = "1) Consultar notas:\n" +
    CheckNotes(ref ps) +
    "2) Abrir láminas: Revisar en caso de mamas. \n" +
    "3) Correspondencia del nombre del paciente, prescripción y
    etapa tratamiento \n" + etapafinal +
    "4) Verificación origen PTs \n" + markersfinal +
    "5) Verificación de dosis de prescripción y número de sesiones
    correcto \n" + prescriptionfinal +
    "6) Correcta posición y mesa de tratamiento \n" +
    positionfinal +
    "7) Técnica de tratamiento adecuada según prescripción \n" +
    tecnicafinal +
    "8) Posición correcta del isocentro (Comprobado en el 9) \n" +
    "9) Todos los campos asociados al mismo acelerador y con el
    mismo isocentro \n" + aceleradorfinal + isocentrofina +
    "10) Matriz de dosis correcta: Comprobar que si técnica de
    tto. es SBRT, resolución = 1,25mm. \n" +
    "11) Verificación algoritmo de cálculo \n" + algoritmofinal +
    "12) Verificación que no hay colisiones con mesa, soportes y/o
    paciente \n" + choquefinal +
    "13) Verificar que se cumplen los objetivos y restricciones:
    En la siguiente. \n" +
    "14) La distribución de dosis correcta por plano, por plan y
    por el tratamiento completo: HACER MANUALMENTE. \n" +
    "15) Verificar que no se exceden los límites de UM de la
    máquina \n" + UMfinal +
    "16) Colocación correcta Bolus: posición, densidad, espesor y
    asignación a haces \n" + bolusfinal +
    "17) (ADICIONAL) Comprobación de que no hay estructuras vacías
    y existe una estructura body de tipo EXTERNAL \n" +
    structuresfinal +
    "\n%%%%%%%%% DETALLADO %%%%%%%%%% \n\n" +
    "3)" + etapa + nombrepaciente +
    "4) " + markers + origin +
    "5) " + prescription +
    "6) " + position + mesa +
    "7) " + tecnica +
    "8 y 9) " + acelerador + isocentro +
    "11) " + algoritmo +
    "12) " + choque +
    "14) La distribución de dosis correcta por plano, por plan y
    por el tratamiento completo no se verifica, hacer esto
    manualmente. \n" +
    "15) " + UM +
    "16) " + bolus +
    "17) " + structures;
```

```
//Segunda ventana emergente (paso 13)
string PrintFinal2 = limites;

MessageBox.Show(PrintFinal1, "PASOS DEL PROCESO");
MessageBox.Show(PrintFinal2, "13) Verificar que se cumplen los
    objetivos y restricciones");
}
}
}
```

---

## Anexo II: Estimación de errores

Los errores vistos en la Tabla 4.1 ha sido obtenido mediante la desviación estándar. Siendo  $\bar{x}$  la media obtenida,  $x$  cada valor independiente y  $N$  el número de valores independientes, el error de  $\bar{x}$  viene dado por:

$$\Delta\bar{x} = \sqrt{\frac{\sum(x - \bar{x})^2}{N}}$$

Para estimar el error en la media de los tres valores, donde cada uno de estos tenía su propio error, se ha procedido mediante propagación de errores. Siendo  $M$  la media de los tres valores  $0.2(x \pm \delta x)$ ,  $0.4(y \pm \delta y)$  y  $0.4(z \pm \delta z)$ , es decir:

$$M = \frac{0.2x + 0.4y + 0.4z}{3}$$