

Trabajo Fin de Máster Máster en Ingeniería Industrial

Diseño de una herramienta computacional
para el cálculo del riesgo de fractura osteopo-
rótica basada en un modelo de remodelación
ósea.

Autor: Paula Gil Gracia

Tutor: Francisco Javier Martínez Reina

**Dep. de Ingeniería Mecánica y Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2023



Trabajo Fin de Máster
Máster en Ingeniería Industrial

Diseño de una herramienta computacional para el cálculo del riesgo de fractura osteoporótica basada en un modelo de remodelación ósea.

Autor:

Paula Gil Gracia

Tutor:

Francisco Javier Martínez Reina

Catedrático de Universidad

Dep. de Ingeniería Mecánica y Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2023

Trabajo Fin de Máster: Diseño de una herramienta computacional para el cálculo del riesgo de fractura osteoporótica basada en un modelo de remodelación ósea.

Autor: Paula Gil Gracia
Tutor: Francisco Javier Martínez Reina

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Resumen

La remodelación ósea es un proceso ejecutado por los huesos que les permite adaptarse a su entorno mecanobiológico, conservando su estructura y funcionalidad. Este proceso les permite mantener su rigidez, resistencia y composición celular óptimas, cumpliendo así sus funciones en el cuerpo humano. Los protagonistas principales de este proceso son los osteoblastos y osteoclastos, células especializadas en la formación y reabsorción ósea, respectivamente. Cualquier desequilibrio en sus actividades puede tener consecuencias graves, como el desarrollo de enfermedades óseas, entre las cuales destaca la osteoporosis.

En este contexto, la ingeniería y la matemática se unen para proporcionar herramientas que permiten analizar, simular y desarrollar tratamientos dirigidos a prevenir o mitigar los efectos de estas enfermedades. Este trabajo se enfoca en la implementación de modelos matemáticos de remodelación ósea interna para el desarrollo de dos interfaces gráficas en App Designer.

Las interfaces desarrolladas tienen como objetivo principal simplificar el estudio de tratamientos para la osteoporosis postmenopausica. La primera proporciona una representación de la evolución de tres parámetros de estudio de la enfermedad: densidad ósea, fracción volumétrica de hueso y daño en tratamientos con denosumab y romosozumab. Además, también facilita la probabilidad de fractura del fémur y vertebra para un tratamiento y condiciones de carga concretos. Mientras que la segunda utiliza redes neuronales para personalizar y optimizar un tratamiento con denosumab.

Abstract

Bone remodeling is a process carried out by bones that allows them to adapt to their mechanical and biological environment, preserving their structure and functionality. This process allows them to maintain their optimal rigidity, resistance and cellular composition, thus fulfilling their functions in the human body. The main protagonists of this process are the osteoblasts and osteoclasts, cells specialized in bone formation and bone resorption, respectively. Any imbalance in their activities can have serious consequences, such as the development of bone diseases, including osteoporosis.

In this context, engineering and mathematics come together to provide tools to analyze, simulate and develop treatments aimed at preventing or mitigating the effects of these diseases. This work focuses on the implementation of mathematical models of internal bone remodeling for the development of two graphical interfaces in App Designer.

The main objective of the interfaces developed is to simplify the study of treatments for postmenopausal osteoporosis. The first one provides a representation of the evolution of three disease study parameters: bone density, bone volume fraction and damage in treatments with denosumab and romosozumab. In addition, it also provides the probability of fracture of the femur and vertebra for a specific treatment and loading conditions. The second uses neural networks to personalize and optimize a denosumab treatment.

Índice Abreviado

<i>Resumen</i>	I
<i>Abstract</i>	III
<i>Índice Abreviado</i>	V
1 Introducción	1
1.1 Justificación del trabajo	1
1.2 Objetivos del trabajo	5
1.3 Estructura del trabajo	6
2 Biología ósea	7
2.1 Tejido óseo	7
2.2 Tipos de huesos	10
2.3 Remodelación ósea	12
2.4 Teoría del mecanostato	14
2.5 Osteoporosis	15
3 Modelo matemático	17
3.1 Ecuaciones diferenciales	17
3.2 Parámetros que describen el modelo	19
3.3 Porosidad en la matriz ósea	23
3.4 Funciones de Hill	23
3.5 Modelo Receptor-Ligando	24
3.6 Inclusión del daño	35
3.7 Inclusión de la mineralización	36
3.8 Osteoporosis (PMO)	38
4 APP DESIGNER: Interfaz gráfica	43
4.1 Introducción a App Designer	43
4.2 Entorno de trabajo	43
4.3 Crear y ejecutar una App sencilla en App Designer	50
5 Calculadora del riesgo de fractura	55
5.1 Justificación y objetivo	55
5.2 Introducción	55

5.3	Datos de entrada y salida	56
5.4	Diseño y desarrollo de la interfaz	57
5.5	Cálculo del parámetro k_5	63
5.6	Implementación de la interfaz	70
6	Redes Neuronales	75
6.1	Introducción a las redes neuronales	75
6.2	Redes neuronales artificiales	75
6.3	Arquitectura de una red neuronal	77
6.4	Métodos de aprendizaje	80
6.5	Funcionamiento de una red neuronal	81
6.6	Implementación de redes neuronales en Matlab	82
6.7	Datos necesarios	84
7	Interfaz optimizadora	87
7.1	Optimización	87
7.2	Interfaz optimizadora	89
7.3	Tratamiento óptimo personalizado	95
7.4	Resultados	103
8	Conclusiones y trabajos futuros	107
	<i>Índice de Figuras</i>	109
	<i>Índice de Tablas</i>	113
	<i>Bibliografía</i>	115

Índice

<i>Resumen</i>	I
<i>Abstract</i>	III
<i>Índice Abreviado</i>	V
1 Introducción	1
1.1 Justificación del trabajo	1
1.1.1 Incidencia de la enfermedad en la población mundial	2
1.1.2 FRAX	3
1.2 Objetivos del trabajo	5
1.3 Estructura del trabajo	6
2 Biología ósea	7
2.1 Tejido óseo	7
2.1.1 Funciones del tejido óseo	7
2.1.2 Composición del tejido óseo	7
Matriz ósea	7
2.1.3 Tipos de tejido óseo	8
Tejido óseo compacto o cortical	8
Tejido óseo esponjoso o trabecular.	8
2.1.4 Células óseas	8
2.2 Tipos de huesos	10
2.3 Remodelación ósea	12
2.3.1 Unidad multicelular básica BMU	12
2.3.2 Remodelación ósea interna	12
Etapas de la remodelación ósea	13
2.4 Teoría del mecanostato	14
2.5 Osteoporosis	15
3 Modelo matemático	17
3.1 Ecuaciones diferenciales	17
3.2 Parámetros que describen el modelo	19
3.3 Porosidad en la matriz ósea	23
3.4 Funciones de Hill	23
3.5 Modelo Receptor-Ligando	24
3.5.1 Señal intracelular y respuesta celular	26

3.5.2	Modelo receptor-ligando competitivo	27
3.5.3	Modelo receptor-ligando competitivo + anticuerpo	30
3.5.4	Regulación del RANKL vía PTH y NO	32
3.5.5	Vía de señalización RANK-RANKL-OPG	33
	RANK	33
	OPG	33
	RANKL	34
3.5.6	Forma de uso de las ecuaciones obtenidas	34
3.6	Inclusión del daño	35
3.7	Inclusión de la mineralización	36
3.7.1	Degradación de las propiedades de la fatiga por mineralización	38
3.8	Osteoporosis (PMO)	38
3.8.1	Inclusión de la edad: Disminución de la liberación de TGF- β de la matriz ósea por reabsorción	38
3.8.2	Inclusión de la edad: Incremento en la producción de esclerostina	39
3.8.3	Inclusión de la Menopausia: Incremento de la producción de RANKL	40
3.8.4	Modelos Pk-PD de Denosumab y Romosozumab	40
4	APP DESIGNER: Interfaz gráfica	43
4.1	Introducción a App Designer	43
4.1.1	Acceso	43
4.2	Entorno de trabajo	43
4.2.1	Elementos de la vista de diseño	44
4.2.2	Elementos de la vista de código	45
4.2.3	Funcionamiento de la vista de código	47
4.2.4	Callbacks, propiedades y funciones	48
4.2.5	Componentes de la librería de componentes	49
4.3	Crear y ejecutar una App sencilla en App Designer	50
5	Calculadora del riesgo de fractura	55
5.1	Justificación y objetivo	55
5.2	Introducción	55
5.3	Datos de entrada y salida	56
5.4	Diseño y desarrollo de la interfaz	57
5.4.1	Diseño en App Designer	57
5.4.2	Puente entre el modelo matemático y la interfaz gráfica	61
5.5	Cálculo del parámetro k_5	63
5.5.1	Introducción	63
5.5.2	Objetivo	64
5.5.3	Matriz 3D	64
	Pasos previos	66
5.5.4	Valor k_5 : obtención e implementación en la interfaz	68
	Obtención del valor k_5	68
	Implementación del k_5 en la interfaz gráfica	69
5.6	Implementación de la interfaz	70
6	Redes Neuronales	75
6.1	Introducción a las redes neuronales	75

6.1.1	Definición de redes neuronales	75
	Machine learning	75
	Redes Neuronales	75
6.2	Redes neuronales artificiales	75
6.2.1	Neurona biológica	75
6.2.2	Neuronas artificiales	75
6.3	Arquitectura de una red neuronal	77
6.3.1	Capa neuronal	77
6.3.2	Clasificación de las redes neuronales	77
	Clasificación por el número de capas:	77
	Clasificación por los tipos de conexión:	78
	Clasificación por grados de conexión:	78
6.3.3	Tipos de redes neuronales	78
6.4	Métodos de aprendizaje	80
6.5	Funcionamiento de una red neuronal	81
6.6	Implementación de redes neuronales en Matlab	82
	Herramientas y paquetes de trabajo	82
	Creación, entrenamiento y evaluación de una red neuronal	82
6.7	Datos necesarios	84
7	Interfaz optimizadora	87
7.1	Optimización	87
7.1.1	Algoritmos de optimización	87
7.1.2	Optimización en Matlab	87
7.2	Interfaz optimizadora	89
7.2.1	Redes neuronales	89
	Obtención de los conjuntos de datos para las redes neuronales	89
	Diseño de las redes neuronales	92
7.3	Tratamiento óptimo personalizado	95
7.3.1	Optimización: Implementación en Matlab	95
	Definición de la función objetivo y sus restricciones	95
7.3.2	Definición de los parámetros de <i>fmincom</i>	98
7.3.3	Optimización: Diseño y desarrollo de la interfaz en App Designer	101
7.4	Resultados	103
8	Conclusiones y trabajos futuros	107
	<i>Índice de Figuras</i>	109
	<i>Índice de Tablas</i>	113
	<i>Bibliografía</i>	115

1 Introducción

Actualmente, debido al progresivo envejecimiento de la población se ha visto incrementado el número de mujeres de tercera edad que padecen osteoporosis postmenopáusica. En este trabajo se presentan dos interfaces gráficas que logran predecir por un lado los datos futuros de una paciente según un tratamiento específico de denosumab y romosozumab y por otro lado un tratamiento óptimo de denosumab personalizado. Las interfaces son sencillas e intuitivas para el usuario, quien necesita proporcionar una serie de datos de entrada característicos de cada paciente. A través de cálculos internos que el usuario no necesita conocer, se obtienen los resultados. Para llevar a cabo este proceso, se ha necesitado modificar el modelo matemático que existe detrás de ambas interfaces para su integración en el entorno de App Designer. Además, para llevar a cabo la optimización se han entrenado una serie de redes neuronales en Matlab que posteriormente se han implementado en un algoritmo de optimización.

1.1 Justificación del trabajo

La osteoporosis es una enfermedad que se caracteriza por una baja densidad mineral ósea y un deterioro de la microarquitectura del tejido óseo. Esto implica una condición en la que el hueso se vuelve más poroso. En consecuencia, se produce un aumento de la fragilidad ósea y la susceptibilidad a fracturas [1].

De hecho, aproximadamente uno de cada ocho europeos mayores de 50 años experimenta una fractura vertebral debido a esta enfermedad, y después de los 80 años, una de cada tres mujeres y uno de cada nueve hombres sufren una fractura de cadera relacionada con la osteoporosis. Por lo tanto, la osteoporosis representa un importante desafío tanto para la salud pública como para la economía, ya que las fracturas causadas por esta enfermedad conllevan una significativa carga socioeconómica.

El resultado clínico de la osteoporosis se manifiesta como fracturas por fragilidad que ocurren en situaciones de traumatismos de baja energía y que no se producirían en circunstancias normales. Las localizaciones más frecuentes de las fracturas son: muñeca, cadera y vértebra.

En la práctica clínica, la osteoporosis se diagnostica principalmente evaluando la densidad mineral ósea (DMO) mediante una prueba llamada absorciometría de rayos X de energía dual. Si se encuentra una puntuación T igual o inferior a -2.5 en la columna lumbar, el cuello del fémur o la cadera, en comparación de la densidad ósea promedio de una población sana de adultos jóvenes.

Sin embargo, no se reduce solo la densidad ósea, existen otros factores que también influyen en el riesgo de fracturas. Es por ello que la mayoría de guías recomiendan que incluso si un paciente no tiene una puntuación T igual o menor a -2.5 debe considerar para el tratamiento de la osteoporosis si existe antecedentes de fractura previa debida a un traumatismo bajo o si su evaluación de riesgo de fractura utilizando la herramienta FRAX muestra un riesgo alto.

FRAX es una herramienta de evaluación del riesgo de fractura que lanzó la Organización Mundial de la Salud (OMS) en colaboración con la universidad de Sheffield, validada científicamente. Esta herramienta calcula la probabilidad a 10 años de fracturas óseas que incluyen otras circunstancias de riesgo además de la baja densidad mineral ósea. Se trata de un hito importante para ayudar a los profesionales sanitarios a mejorar la identificación de los pacientes con alto riesgo de fractura

1.1.1 Incidencia de la enfermedad en la población mundial

Los siguientes datos pueden resultar interesantes para poner de manifiesto la incidencia que tiene la enfermedad a nivel mundial.

- En todo el mundo, la osteoporosis causa más de 8,9 millones de fracturas al año [2].
- Según la OMS, la enfermedad afecta aproximadamente al 6,3% de los hombres mayores de 50 años y al 21,2% de las mujeres del mismo rango de edad a nivel mundial. Esto sugiere aproximadamente que 500 millones de hombres y mujeres en todo el mundo pueden verse afectados [3].
- Se estima que en el año 2000 ocurrieron aproximadamente 9 millones de fracturas nuevas debido a la osteoporosis. De estas, 1.6 millones afectaron a la cadera, 1.7 millones al antebrazo y 1.4 millones fueron fracturas vertebrales clínicas. La mayor proporción de estas fracturas, alrededor del 51%, ocurrió en Europa y América. El resto se distribuyó principalmente en la región del Pacífico occidental y el sudeste asiático [2].
- En toda Europa en 2019 (Unión Europea más Suiza y Reino Unido), se estima que 32 millones de personas mayores de 50 años tenían osteoporosis, lo que da lugar a un 5,6% de la población Europea total de más de 50 años [4].

Las fracturas debidas a la osteoporosis en mujeres mayores de 45 años tienen un impacto significativo en la salud y la calidad de vida [5]:

- Estas fracturas requieren más días de hospitalización que muchas otras enfermedades, como la diabetes, los infartos y el cáncer de mama.
- Hasta el 20-24% de los pacientes fallecen en el primer año después de una fractura de cadera.
- Los sobrevivientes de fracturas de cadera experimentan pérdida de independencia, con un 40% incapaces de caminar de forma independiente y un 60% que necesita asistencia un año después.
- El 80% de ellos enfrenta restricciones en otras actividades, como conducir y hacer compras.
- Un tercio de los pacientes con fractura de cadera se vuelven totalmente dependientes o necesitan cuidados en una residencia de ancianos durante el año posterior a la fractura.
- Las fracturas no solo afectan físicamente, sino también emocionalmente, causando depresión y aislamiento.
- La pérdida a largo plazo de independencia y movilidad genera tensiones físicas, emocionales y financieras tanto para los pacientes como para sus familiares y amigos.

Según un informe reciente de la Fundación Internacional de Osteoporosis un total de 2.945.000 personas padecen osteoporosis en España. El 79,2% de las personas que la padecen son mujeres y el 30% son hombres. Además, se estima que el número de fracturas por fragilidad crecerá aproximadamente un 30% entre 2019 y 2024. En España se produjeron 289.000 fracturas por fragilidad en 2019, que es equivalente a 33 fracturas por día [6].

El coste de las fracturas osteoporóticas (ver figura 1.1) contabilizado en España supone aproximadamente el 3,8% del coste sanitario (es decir, 4.300 millones de euros de 104.300 millones en 2019, algo más que el 3,5% de la Unión Europea) [6]. El coste económico de las fracturas por fragilidad, 4,3 miles de millones en 2019, equivalente a 92,3 euros por persona, supone un aumento del 33% en 2010 [6].

Las fracturas de cadera y columna vertebral en adultos mayores son una causa importante de morbilidad y mortalidad.



Figura 1.1 Coste de la enfermedad. Tomada de [7].

1.1.2 FRAX

El FRAX, una herramienta desarrollada por el Centro Colaborador de la Organización Mundial de la Salud (OMS) en Sheffield, Reino Unido, en 2008, ha desempeñado un papel significativo en la gestión de la osteoporosis. Es una herramienta online (ver figura 1.2), que permite calcular el riesgo de fractura a partir de una serie de factores de riesgos clínicos, tanto si se conoce la densidad mineral ósea como si no. Este permite calcular el riesgo de fractura ósea en los próximos 10 años tanto en términos generales como concretamente en la cadera. Se enfoca en las personas con edades entre 40-90 años y que no reciben tratamiento para la osteoporosis. Los modelos FRAX han sido desarrollados a partir de estudios poblacionales, representando poblaciones de riesgo elevado (Suecia y Estados Unidos), riesgo moderado (China, España, Japón y Francia) y riesgo bajo (Turquía). El FRAX ha sido valorado durante mucho tiempo por los profesionales de la salud por su aplicabilidad en la atención primaria y su simplicidad. A pesar de esto, también ha sido criticada por académicos por las mismas razones [8].

El proceso de creación del FRAX implicó la recopilación y el análisis de datos de estudios de cohortes a nivel mundial que evaluaron los factores de riesgo asociados con las fracturas osteoporóticas. La información proporcionada por las cohortes se integró en una regresión de Poisson en la que los eventos de fractura y muerte fueron incluidos como funciones continuas [8].

Estas son las variables que tiene en cuenta el FRAX:

- Edad
- Sexo
- Índice de masa corporal
- Antecedente de fractura por fragilidad en la edad adulta
- Antecedente de fractura de la cadera en alguno de los progenitores

- Tabaquismo activo
- Ingesta de glucocorticoides
- Artritis reumatoide
- Osteoporosis secundaria
- Consumo excesivo de alcohol
- Densidad mineral ósea

Herramienta de Cálculo

Por favor responda las preguntas siguientes para calcular la probabilidad de fractura a diez años sin DMO o con DMO.

país: **Reino Unido** Nombre/ID: [Sobre los Factores de riesgo](#)

Cuestionario:

1. Edad (entre 40-90 años) o fecha de nacimiento
Edad: Fecha de Nacimiento: A: M: D:

2. Sexo Hombre Mujer

3. Peso (kg)

4. Estatura (cm)

5. Fractura previa No Sí

6. Padres con Fractura de Cadera No Sí

7. Fumador Activo No Sí

8. Glucocorticoides No Sí

9. Artritis Reumatoide No Sí

10. Osteoporosis secundaria No Sí

11. Alcohol, 3 o más dosis por día No Sí

12. DMO de Cuello Femoral
Seleccione BMD

[Herramienta de impresión y la información](#)

Figura 1.2 Web del FRAX [9].

A continuación, se exponen algunas de las ventajas y desventajas de esta herramienta:

Ventajas:

- **Rigor metodológico:** Al utilizar datos recolectados directamente de los cohortes, mejora la precisión en el cálculo del riesgo de fractura al permitir la interacción de factores de riesgo con el riesgo de fractura. Esto es importante en variables continuas como la edad, la densidad mineral ósea (DMO) y el índice de masa corporal. Además, el uso de la regresión de Poisson en el FRAX permite evaluar cuándo ocurren fracturas y muertes de manera continua en el tiempo, considerando también el impacto de los factores de riesgo de fractura en la mortalidad [8].
- **Utilización del riesgo absoluto:** El riesgo absoluto de fractura representa la probabilidad de sufrir una fractura en un período de tiempo específico, lo que facilita su comprensión tanto para médicos como para pacientes. A menudo, los ensayos clínicos presentan resultados en términos de reducción del riesgo relativo, que compara la frecuencia de fracturas entre aquellos con un factor de riesgo o que toman un medicamento y aquellos que no lo hacen. Hablar en términos de riesgo absoluto transforma la información en algo más comprensible. La probabilidad a 10 años se vuelve más relevante para tomar decisiones terapéuticas [8].

- **Facilidad en atención primaria:** En la práctica clínica diaria, el FRAX ayuda a orientar a los pacientes, siendo un soporte para presentar los datos a cerca del riesgo de fractura de una manera fácil y entendible. Dando así una estimación del riesgo sencilla para a partir de ella tomar decisiones [8].

Desventajas:

Las limitaciones del FRAX tienen su origen principalmente en la falta de calidad en la recopilación de datos en las bases de datos utilizadas para su cálculo:

- **Impacto de la Exclusión en Estudios de Cohortes:** En los estudios de cohortes, a veces ocurre que algunas personas no quieren o no pueden participar en el estudio. Esto puede llevar a que las personas más enfermas o con más problemas de salud no estén representadas en la investigación. En la osteoporosis, esto podría significar que las personas que tienen un mayor riesgo de fractura, pero que también tienen más problemas de salud, no estén incluidas en el estudio. Como resultado, el riesgo de fractura estimado en el estudio podría ser más bajo de lo que sería en la población general, ya que no se están teniendo en cuenta las personas más enfermas que tienen un mayor riesgo.
- **Peor escenario sin DMO:** Su funcionamiento es mejor si se conoce la DMO. Si no se conoce, por lo general el riesgo de fractura predicho es mayor.
- **Ausencia de consideración de la dosis en los factores de riesgo:** No tiene en cuenta la respuesta a las dosis de varios factores de riesgo. El FRAX no considera la cantidad o intensidad de varios factores de riesgo que afectan al riesgo de fractura. Simplemente clasifica los factores de riesgo en categorías de "SÍ" o "NO", sin tener en cuenta la magnitud del riesgo adicional que pueden representar.
- **Omisión de factores relevantes:** No están considerados todos los factores de riesgo conocidos para fracturas debido a la falta de datos adecuados en las bases de datos utilizadas. Algunos de estos factores incluyen deficiencia de vitamina D, historial de caídas, nivel de actividad física, indicadores del metabolismo óseo, historial de tratamientos previos para la osteoporosis, etc.
- **Falta de diferenciación de la ubicación de las fracturas previas.** EL FRAX considera las fracturas previas pero no distingue entre los diferentes sitios. Esto significa que una fractura previa en una parte del cuerpo que no está directamente relacionada con el riesgo de fractura de cadera o columna vertebral se tendrá en cuenta de la misma manera que una fractura previa en una ubicación de alto riesgo.

Como conclusión, a pesar de que esta herramienta tiene utilidad en su uso en Atención primaria, no debe considerarse como el único factor determinante para tomar decisiones terapéuticas.

1.2 Objetivos del trabajo

Dada la relevancia de esta enfermedad, en este trabajo se propone abordar dos objetivos fundamentales:

En primer lugar, se dispone de un modelo matemático que simula diversos tratamientos para la osteoporosis postmenopáusica en pacientes concretos. La manipulación de este modelo resulta compleja debido a su extensión y el gran número de ecuaciones que involucra. Por consiguiente, se plantea en este trabajo la necesidad de adaptar este modelo, a través de una interfaz gráfica, con el fin de simplificar la representación de los datos esenciales para su uso. Esto posibilita la utilización del modelo de manera sencilla sin necesidad de tener conocimiento sobre el.

Por otro lado, el denosumab es un medicamento utilizado en el tratamiento de la osteoporosis postmenopáusica. Su objetivo recae en reducir el riesgo de fracturas óseas. Su mecanismo de acción

se centra en inhibir la reabsorción ósea, ayudando a mantener una densidad ósea equilibrada. Las inyecciones de denosumab se administran generalmente una vez cada seis meses en una dosis de 60 mg. No obstante, esta cantidad puede resultar insuficiente para ciertas pacientes o excesiva para otras. Es por esta razón que se plantea la necesidad de adaptar el tratamiento de manera individualizada para cada persona. En este trabajo se ajustan las variables dosis y lapsos entre inyecciones con el fin de diseñar un tratamiento específico y personalizado para cada paciente.

1.3 Estructura del trabajo

Este trabajo se estructura en base a capítulos. Tras este primero de introducción, donde se ha justificado el trabajo y se han presentado los objetivos, en el segundo capítulo, se aborda la biología ósea, que comprende los tipos de huesos, sus funciones y composición. También se explica la remodelación ósea, un aspecto fundamental que ejerce una influencia significativa en la enfermedad objeto de estudio, la osteoporosis postmenopáusica, la cual se explica también en este mismo capítulo.

En el tercer capítulo, se presenta en profundidad el modelo matemático empleado para simular el hueso, tomando en consideración la remodelación ósea, la enfermedad y sus tratamientos.

En el cuarto capítulo, se aborda la herramienta de desarrollo App Designer. Se proporciona una explicación detallada acerca de esta herramienta, se analizan sus componentes principales y se presenta un ejemplo de su aplicación.

En el capítulo quinto, una vez que se conoce App Designer, se procede a detallar el proceso de desarrollo de la primera interfaz, denominada 'Calculadora del Riesgo de Fractura'. A través de esta interfaz, se hace posible la obtención de resultados relacionados con el tratamiento de la osteoporosis mediante el ingreso de los datos de la paciente en la interfaz.

En el sexto capítulo, se aborda el conocimiento de las redes neuronales, se detallan sus características principales y clases, y se proporciona una explicación sobre su implementación Matlab.

En el séptimo capítulo, se introduce y detalla la segunda interfaz denominada "Interfaz Optimizadora". Se inicia con la definición del concepto de optimización y posteriormente se explica el proceso llevado a cabo mediante Matlab y App Designer para lograr un tratamiento óptimo basado en las redes neuronales previamente presentadas. Este proceso da lugar a la creación de una interfaz gráfica que proporciona el tratamiento óptimo específico para cada paciente gracias a la implementación de las redes neuronales y la función *fmincom* de Matlab.

Finalmente en el capítulo 8 se presentan las conclusiones y se esbozan algunas mejoras del modelo y de la herramienta que se pueden abordar en trabajos futuros.

2 Biología ósea

El sistema esquelético, también conocido como sistema óseo, es uno de los sistemas principales del cuerpo humano. Los huesos se conectan entre sí a través de articulaciones para formar el esqueleto y están compuestos por varios tipos de tejidos, siendo el tejido óseo el más predominante. La misión principal del sistema esquelético es conformar la estructura del cuerpo y dar soporte. El sistema esquelético debe adaptarse constantemente a las cargas a las que se expone a diario. Esto se logra a través de un proceso llamado remodelación ósea, mediante el cual los huesos se adaptan a las demandas de cargas a las que se encuentran sometidos. En este capítulo se van a introducir el tejido óseo, la estructura de los huesos y el proceso de remodelación ósea.

2.1 Tejido óseo

2.1.1 Funciones del tejido óseo

El tejido óseo es un tipo de tejido conectivo que se caracteriza por su rigidez y su gran resistencia a la tracción como a la compresión. Entre sus diversas funciones destacan dos: proporcionar soporte interno al organismo y la protección de los órganos vitales contra impactos. Toma un papel importante en la generación de movimiento, ya que posibilita este gracias a la interacción entre músculos, articulaciones y huesos. Otras funciones significativas son el almacenaje de minerales como el fósforo y el calcio, necesarios para el equilibrio mineral del cuerpo y albergar la médula ósea, fuente principal de células madre y células sanguíneas [10]

2.1.2 Composición del tejido óseo

El tejido óseo es un componente vivo, formado por células y una matriz ósea, ver figura 2.1 [11]:

Matriz ósea

- Fase inorgánica mineral: Representa el 70% de la masa ósea. Está formada por fosfato de calcio ($\text{Ca}_{10}(\text{PO}_4)_6(\text{OH})_2$) que forma los cristales de hidroxiapatita que le proporciona resistencia y dureza al tejido óseo.
- Fase orgánica: Está formada principalmente de colágeno tipo I, aunque también se encuentra el tipo III y el VI. También se encuentran otras proteínas no colágenas como la osteocalcina, la osteonectina o la osteopontina. Representa aproximadamente el 20% de la masa total del hueso
- Agua: representa aproximadamente el 10% restante.



Figura 2.1 Composición de la matriz ósea..

2.1.3 Tipos de tejido óseo

Existen diferentes formas de describir la organización del tejido óseo. A continuación se presentará una de ellas a nivel macroscópico. En esta descripción, el hueso se compone de dos tipos de tejido que difieren en su porosidad:

Tejido óseo compacto o cortical

Se compone del 80% del esqueleto humano. Su matriz ósea es de alta densidad y tiene poca porosidad, lo que significa que no hay muchas cavidades en ella. Está formado por una variedad de láminas y canales. Los vasos sanguíneos que alimentan el hueso y los nervios pasan por el conducto de Havers a través de las láminas concéntricas. La osteona es el conjunto formado por el conducto de Havers y las laminillas óseas con osteocitos. Además, los canales de Havers están conectados por canales transversales llamados Volkman. La superficie central (diáfisis) de los huesos contiene tejido compacto [10]

Tejido óseo esponjoso o trabecular.

Este hueso es de porosidad alta y se encuentra en la epífisis de los huesos (los huesos cortos, en los huesos planos y en los extremos de los huesos largos). Está formado, ver figura 2.2, por una red tridimensional de trabéculas óseas ramificadas, dispuestas de manera ordenada o entrecruzadas que limitan un laberinto de espacios interconectados ocupados por la medula ósea [10]

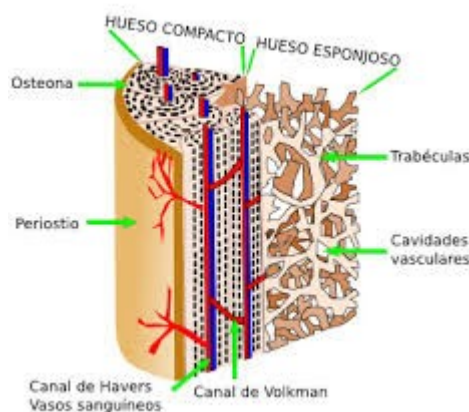


Figura 2.2 Estructura del hueso a nivel macroscópico. Tomada de [12].

2.1.4 Células óseas

En el tejido óseo se encuentran cuatro tipos de células especializadas: osteocitos, osteoblastos, osteoclastos y células de borde. Los osteoblastos son los responsables de la formación del nuevo

tejido óseo, por el contrario, los osteoclastos son los encargados de realizar la reabsorción ósea. Juntos regulan el crecimiento óseo.

Las células osteoprogenitoras o células madre del hueso son células que actúan como progenitoras del hueso. Se ubican sobre la superficie esperando a ser activadas por los estímulos hormonales. Al activarse se convierten en osteoblastos. Los osteoclastos, por otro lado, derivan de células hematopoyéticas. Finalmente, los osteocitos y células de borde se diferencian de los osteoblastos para desempeñar una función distinta.

- **Osteoclastos:** Su trabajo es realizar la reabsorción del hueso, ver figura 2.3. Debido a que son células multinucleadas con alrededor de 10 a 20 núcleos, poseen un gran tamaño. Proviene de la fusión de los osteoclastos precursores. Tras esto, múltiples osteoclastos maduros se activan simultáneamente, produciendo ácidos y enzimas que disuelven la matriz del hueso. Estas células mueren por muerte programada o apoptosis una vez que han completado su función. Tras la acción de los osteoclastos aparece un hueco en la matriz ósea.



Figura 2.3 Osteoclasto reabsorbiendo superficie ósea. Tomada de [13].

- **Osteoblastos:** Su función es formar el nuevo tejido óseo. Son células mononucleadas y pequeñas que se ubican en la superficie del hueso y en áreas donde se necesita formar nuevo tejido óseo, ver figura 2.4. Proceden de células madre mesenquimales y pasan por diferentes etapas de diferenciación, siendo la última la formación de osteocitos. Para su diferenciación requieren cierto nivel de estímulo mecánico. Una vez que han cumplido su función, pueden experimentar apoptosis o convertirse en osteocitos o células de borde.

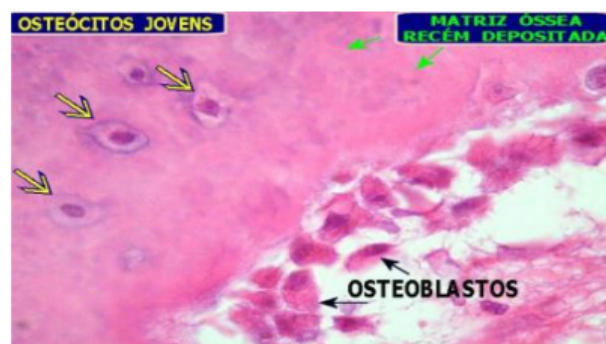


Figura 2.4 Osteoblastos. Tomada de [13].

- **Osteocitos:** Son las células óseas más comunes. Los osteoblastos se separan y se convierten en osteocitos una vez que se mineraliza la matriz. Cuando esto sucede, el osteocito desarrolla prolongaciones que le dan la forma de una estrella. La célula central y sus prolongaciones se

encuentran en pequeñas lagunas conectadas por canalículos (ver figura 2.5). Estos pueden comunicarse entre sí usando estas conexiones. Su función consiste en el mantenimiento del hueso, asegurándose de que este está soportando una carga adecuada, si esto no es así, envía señales a las células de borde para comenzar la remodelación.

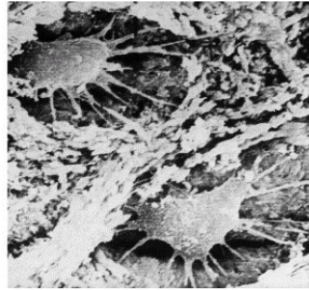


Figura 2.5 Osteocitos conectados mediante sus prolongaciones. Tomada de [14].

- **Células de borde:** Estas células, ver figura 2.6, desempeñan un papel fundamental en la protección de la matriz ósea contra los estímulos químicos que podrían disolverla. Las células de borde están conectadas a los osteocitos a través de los canalículos y cuando estos detectan una anomalía en la matriz ósea las células de borde son las encargadas de enviar las señales químicas que activan el proceso de remodelación. Una vez comienza la remodelación, las células de borde se retiran de la superficie del hueso.

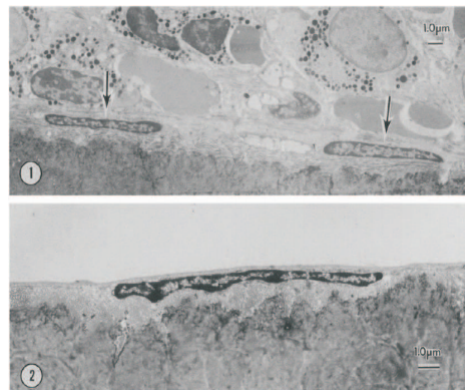


Figura 2.6 Células de borde.

2.2 Tipos de huesos

Los huesos pueden clasificarse (ver figura 2.8) según su forma en:

- **Huesos largos:** La función principal de los huesos largos es proporcionar soporte estructural al cuerpo, resistencia y facilidad en los movimientos. Estos huesos están diseñados para soportar cargas y transmitirlas a través de su estructura. Entre ellos se encuentra el fémur y el radio. En ellos se pueden distinguir tres regiones anatómicas (figura 2.7):
 - La epífisis, que son los dos extremos y está formada por hueso trabecular.
 - La diáfisis, que es un cilindro hueco de paredes gruesas formadas por tejido óseo compacto.

- La metáfisis es la zona del hueso donde la diáfisis se ensancha en los extremos.

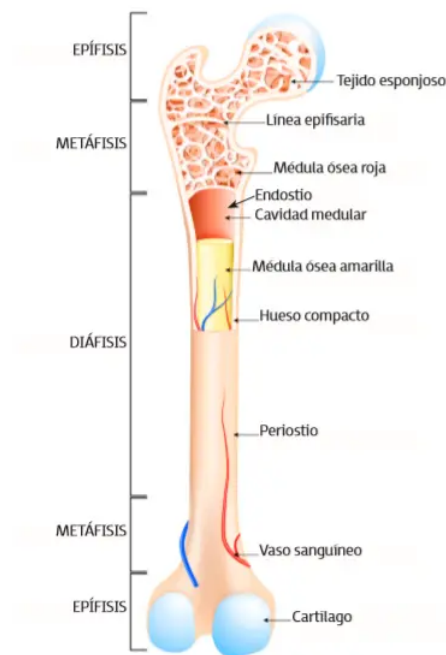


Figura 2.7 Partes del hueso largo. Tomada de [15].

- **Huesos cortos:** Tienen aproximadamente la misma longitud que ancho. Su principal función es dar sostén y soportar pesos. Están compuestos por una capa externa de tejido óseo compacto y una zona central de hueso esponjoso. Son los encargados de permitir el movimiento de las extremidades y brindar estabilidad. Entre ellos el carpo (huesos de la mano) y el tarso (huesos del pie).
- **Huesos planos:** Su función principal es la protección de los órganos internos. A diferencia de otros huesos, apenas soportan cargas. Están formados por una capa de hueso esponjoso entre dos capas delgadas de hueso compacto. Su característica distintiva es su forma aplanada. Entre ellos destaca el cráneo.
- **Huesos irregulares:** Representan todos aquellos huesos que por su forma no se pueden clasificar en otro tipo. Con frecuencia tienen formas bastante complejas, como los huesos que conforman el oído.
- **Huesos sesamoideos:** Son huesos que están incluidos en tendones. Su función es proteger a los tendones del estrés y el deterioro.

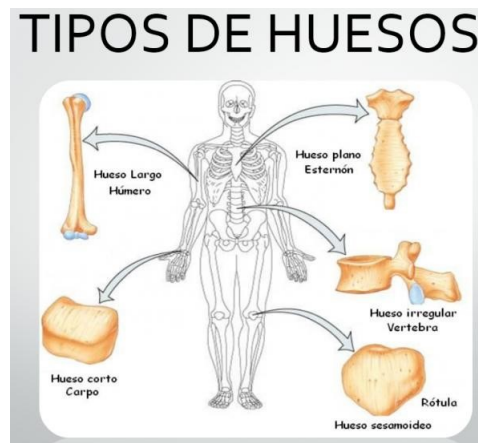


Figura 2.8 Ejemplos de los distintos tipos de huesos..

2.3 Remodelación ósea

El proceso de remodelación ósea consiste en la adaptación del hueso en función de las cargas que soporta. Implica la eliminación del tejido óseo dañado y la formación de nuevas células que reemplacen y desempeñen las funciones del sistema esquelético. Este mecanismo permite mantener la salud del tejido óseo y reparar y remodelar los huesos en función de las posibles lesiones o cambios sufridos por las cargas aplicadas en ellos [16].

Se distinguen dos tipos de remodelación ósea (RO), la remodelación ósea interna (ROI) y la remodelación ósea externa (ROE). En la ROI los cambios se dan en el interior del hueso y ocurren de forma continua durante toda la vida, mientras que, en la ROE los cambios se dan en el exterior del hueso y se dan en momentos específicos, como la etapa de crecimiento o en la senectud. Este trabajo se ha centrado en la ROI. Las células óseas que intervienen en la RO son las que conforman el tejido óseo: Osteoclastos, Osteoblastos, Osteocitos y Células de borde.

2.3.1 Unidad multicelular básica BMU

Unidad multicelular básica son las siglas de BMU en inglés 'Basic Multicellular Unit' [17], es una población de células formada por varios grupos de células que trabajan conjuntamente para llevar a cabo la remodelación ósea. Está formada por osteoblastos, osteocitos, osteoclastos y células de borde. El equilibrio entre la reabsorción y la formación ósea, se denomina homeostasis, el número de células necesarias está regulada por esta. Por lo que el propósito de las BMU es mantener el equilibrio y la homeostasis del tejido óseo. Esta realiza de forma secuencial y organizada el proceso de remodelación ósea.

2.3.2 Remodelación ósea interna

La ROI tiene lugar durante toda la vida. La función principal de esta es mantener el tejido y preservar su estructura óptima. Esta remodelación no afecta ni al tamaño ni a la forma del tejido, además ocurre de manera circunstancial, diferenciándose un claro inicio y final. Esta remodelación se lleva a cabo gracias al mecanismo BMU, desarrollado anteriormente en el punto 1.3.1. La BMU se caracteriza por la acción coordinada de las células óseas, osteoblastos y osteoclastos, lo que es imprescindible para la ROI. Cada BMU está compuesta por osteoclastos y por una gran cantidad de osteoblastos. Organizados de una forma secuencial y con una actividad ordenada y sucesiva. Esta secuencia de operaciones es conocida como ARF (Activación-Reabsorción-Formación) [18]. Cada ciclo de esta consta de seis etapas a desarrollar en el siguiente punto.

Etapas de la remodelación ósea

Durante el proceso de remodelación ósea se suceden una serie de fases, ver figura 2.9, que van llevando a cabo la reabsorción del hueso y posteriormente la formación. Sin embargo, existen otras etapas que también son importantes; activación, reabsorción, inversión, formación, mineralización y final. A continuación, se desarrollan estas etapas:

- Activación

El tejido óseo está compuesto por una red de osteocitos conectados entre sí, y estos a su vez a las células de borde por medio de ramificaciones que atraviesan los poros. Por estos huecos fluye el riego sanguíneo que proporciona al tejido nutrientes. Los osteocitos toman un papel fundamental en esta etapa, ya que estos son muy sensibles a los esfuerzos tangenciales, o lo que es lo mismo, ellos 'sienten' la carga mecánica.

El hueso se deforma como resultado de las cargas, lo que provoca diferencias de presiones en los canalículos dando lugar a un flujo de nutrientes que para los osteocitos son esfuerzos tangenciales. Según la Teoría inhibitoria de Martin [19], si el flujo se altera en caso de desuso o sobrecarga, los osteocitos dejarán de sentir las tensiones tangenciales y dejarán de transmitir una señal bioquímica a las células de borde que impide que se active la actuación de la BMU.

- Reabsorción

Según la teoría inhibitoria, la tendencia de las células de borde es promover la remodelación ósea en todo momento, pero la señal inhibitoria que se recibe de los osteocitos se lo impide. Por ello, si a las células de borde dejan de llegarles la señal de los osteocitos, da comienzo la remodelación ósea. En ese momento, comienzan a segregar una sustancia química denominada RANKL, que es detectada por la proteína RANK de los precursores de los osteoblastos presentes en la médula ósea provocando su diferenciación y fusión en los osteoclastos maduros. Simultáneamente, las células de borde cambian su forma contrayéndose, para dejar libre la superficie del hueso que van a absorber los osteoclastos. Estas células reabsorben el tejido óseo, disuelven la matriz extracelular, solubilizando primero el mineral y luego dirigiendo la matriz orgánica. Una vez finalizada su función, sufren apoptosis. La reabsorción se da de formas distintas según el tipo de hueso; si es cortical se origina un túnel circular mientras si es trabecular es una cavidad superficial.

- Inversión

En esta fase se produce el cambio de actividad entre los osteoclastos y los osteoblastos. La activación de los últimos para formar el nuevo tejido óseo no es inmediata. En el ser humano transcurren alrededor de 30 días desde la activación de los osteoclastos. Se caracteriza por su aparente inactividad.

- Formación

Los osteoblastos se posicionan sobre las cavidades originadas por los osteoclastos y las rellenan con lamelas de osteoide, formadas de agua y colágeno. Una vez realizada su función los osteoblastos se diferencian en osteocitos o células de borde o sufren apoptosis.

- Mineralización

Durante la etapa de formación ósea los osteoblastos depositan osteoide que no está compuesto por minerales, por ello es necesaria una posterior mineralización. Esta se inicia en la interfase entre el osteoide y el hueso, avanzando hacia el exterior. A medida que avanza va dejando una matriz ósea ya mineralizada en forma de cristales de hidroxiapatita. Esta mineralización consta de dos fases:

- **Fase primaria:** Fase muy rápida en la que se llega al 70% de contenido mineral.

- **Fase secundaria:** Fase más lenta con velocidad de deposición que decrece exponencialmente en la que se llega al 95% de contenido mineral.

- **Final**

Para acabar, los osteoblastos sufren apoptosis y la BMU se detiene. Los osteoblastos terminarán de rellenar la cavidad y se diferenciarán a osteocitos o células de borde. Aunque la BMU se detenga, la mineralización continua.

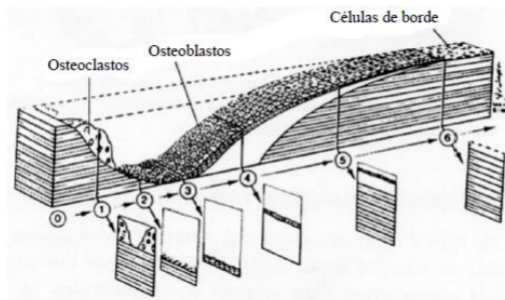


Figura 2.9 Fases de la actividad de una BMU de un hueso trabecular: (0) Activación, (1) Reabsorción, (2) Inversión, (3) Comienzo de la formación, (4) Formación de osteodide, (5) Mineralización, (6) Final. Tomada de [20].

2.4 Teoría del mecanostato

La teoría del mecanostato, ver figura 2.10, fue propuesta por Frost [21] y según esta, el tejido óseo posee un mecanismo intrínseco que él denominó 'mecanostato'. Su funcionamiento es similar al de un termostato que regula la temperatura, pero en el caso del tejido óseo la variable a regular es la densidad ósea. La velocidad con la que varía la temperatura dentro de una habitación es proporcional a la diferencia entre la temperatura de la estancia y la temperatura de control del termostato. De la misma forma, la velocidad con la que varía la densidad ósea o la velocidad de remodelación ósea es proporcional a la diferencia entre la deformación que soporta el tejido y una deformación de referencia, que es la misma para cualquier tipo de hueso.

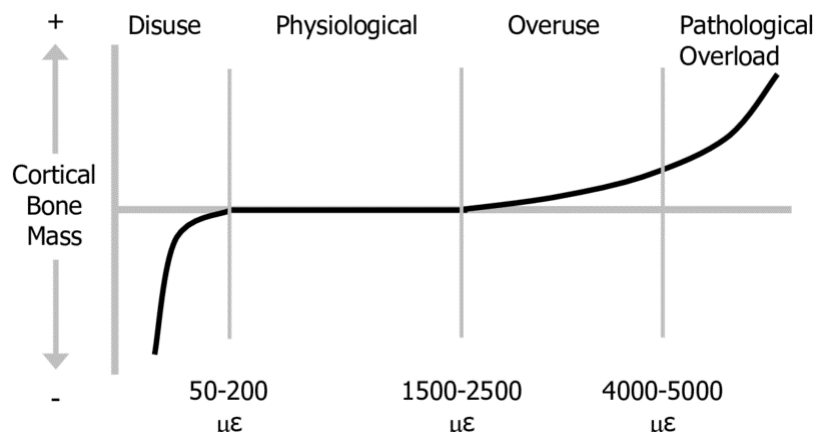


Figura 2.10 Teoría del mecanostato de Frost. Adaptación del hueso a estímulos mecánicos. Tomada de [22].

2.5 Osteoporosis

La osteoporosis es una enfermedad que se caracteriza por una disminución de la densidad de los huesos a causa de la pérdida del tejido óseo normal [23]. Como resultado, los huesos se vuelven más frágiles y susceptibles a fracturas debido a traumatismos o cargas, ver figura 2.11.

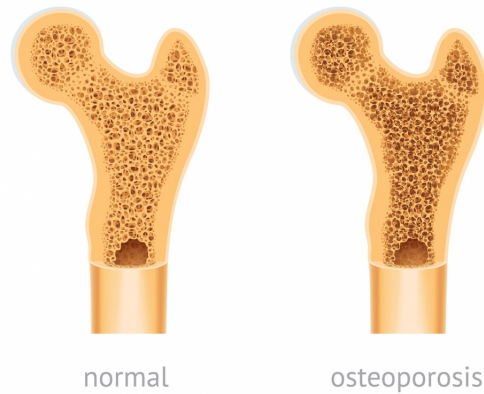


Figura 2.11 Diferencia de porosidad en hueso sano y enfermo. Tomada de [24].

El tejido del hueso se encuentra en constante formación y reabsorción. Cuando el equilibrio entre ambas se ve interferido aparece la osteoporosis. Puede darse por la disminución de de formación, por el aumento de reabsorción o por ambas a la vez. En esta condición, los huesos se vuelven más porosos, lo que implica un aumento en el número y tamaño de las cavidades presentes en su interior. Como resultado, los huesos se vuelven más delgados y frágiles, lo que los hace más propensos a sufrir fracturas incluso ante traumatismos de menor impacto.

Las fracturas debidas a la osteoporosis son una consecuencia común y se pueden dar en diferentes partes del cuerpo. Las zonas más frecuentes son la muñeca, la cadera y las vertebras, ver figura 2.12 [25].



Figura 2.12 Fracturas típicas a causa de la osteoporosis [26].

Existe dos tipos generales de osteoporosis la primaria y la secundaria [27]:

- Osteoporosis primaria:
 - Osteoporosis idiopática juvenil o del adulto:

- Osteoporosis tipo I o postmenopausica: Se da en mujeres después de la menopausia debido a la reducción de estrógenos, hormonas que desempeñan un papel importante en la salud ósea.
- Osteoporosis tipo II o senil: Producida por el envejecimiento.
- Osteoporosis secundaria: Es aquella causada por otras patologías o por la ingesta de ciertos medicamentos.

Esta enfermedad puede afectar a cualquier persona, aunque existen algunos factores de riesgo:

- **Sexo.** Las mujeres son más propensas a desarrollar osteoporosis que los hombres.
- **Edad.** El aumento de edad conlleva un aumento del riesgo de osteoporosis.
- **Raza.** Más tendencia a sufrirla la raza blanca que la negra o asiática.
- **Antecedentes familiares.** Si existen antecedentes familiares de osteoporosis. Para una persona aumenta el riesgo de desarrollarla si uno de sus padres o hermanos han sido diagnosticados.
- **Tamaño del cuerpo.** Las personas con estructuras corporales pequeñas tienen más posibilidades de sufrirla al tener menor masa ósea.

Existe una relación directa entre la falta de estrógenos después de la menopausia y el desarrollo de la osteoporosis [28]. Tras la menopausia la reabsorción aumenta mientras que la generación de hueso disminuye. Esto se debe a que la disminución de los estrógenos conlleva un aumento de los osteoclastos y con ello un aumento de la reabsorción. En este trabajo, se ha tratado la osteoporosis postmenopáusicas.

Para tratar la osteoporosis es necesario de manera general seguir una dieta rica en calcio y baja en proteínas, realizar ejercicio físico de manera habitual y evitar el consumo de tabaco y alcohol. Existen diversos tratamientos farmacológicos, en este trabajo destacan dos:

- El **denosumab** es un fármaco del tipo denominado anticatabólico, es decir que se administra para disminuir la reabsorción del hueso, que se puede descontrolar en mujeres posmenopáusicas tras la bajada de estrógenos asociada a la menopausia. Es un anticuerpo monoclonal humano inhibidor del ligando receptor activador de RANKL. Este fármaco actúa uniéndose al RANKL y bloqueando su acción, evitando el aumento de osteoclastos, con ello disminuyendo la reabsorción y por tanto, la pérdida de masa ósea. La administración de este fármaco se realiza mediante inyecciones subcutáneas cada seis meses y ha resultado ser efectivo en la reducción del riesgo de fractura, a pesar de algunos efectos secundarios. En este trabajo se plantea encontrar un tratamiento con este fármaco específico de paciente [29].
- El **romosozumab** presenta un mecanismo dual sobre la remodelación ósea. Por un lado incrementa la formación de hueso potenciando la proliferación de células precursoras de osteoblasto. Por otro lado disminuye la reabsorción ósea. En su conjunto, esta doble acción, que implica el incremento de la formación ósea y la reducción de la reabsorción ósea, conduce a un rápido aumento de la densidad mineral tanto en el tejido óseo trabecular como en el cortical. Esto se traduce en mejoras notables en la estructura ósea y en su capacidad de resistencia. La dosis aprobada por la OMS es de 210 mg de romosozumab (administrados como dos inyecciones por vía subcutánea de 105 mg cada una) una vez al mes durante doce meses [30].

Además del tratamiento personalizado de denosumab, este trabajo también se propone evaluar la densidad ósea y el riesgo de fractura en un individuo, a través de una interfaz gráfica, en el caso de que se le administre cualquiera de estos dos tratamientos.

3 Modelo matemático

Dentro de la variedad de modelos matemáticos que existen en la literatura para simular la remodelación ósea interna, se pueden distinguir dos categorías principales: los fenomenológicos y los mecanobiológicos.

Los modelos fenomenológicos se pueden denominar 'caja negra', estos establecen relaciones matemáticas entre las variables de entrada y las variables de salida. Las variables de entrada suelen ser las cargas mecánicas soportadas por el hueso y las variables de salida la densidad ósea y las propiedades mecánicas del hueso, como la rigidez [31, 32]. Una aplicación típica de estos modelos es la estimación de la distribución de densidad ósea en el hueso a partir de la cargas mecánicas que soporta [33]. Otra aplicación es la variación de densidad que produce un cambio en el hueso como la introducción de un implante [34].

A diferencia de los modelos fenomenológicos, los modelos mecanobiológicos intentan describir procesos biológicos más detallados que tiene lugar dentro del hueso. Estos modelos consideran las concentraciones de células y factores bioquímicos. Así logran obtener información de lo que ocurre y no solo de los cambios que se producen en las propiedades mecánicas [35, 36]. Un tipo de modelo mecanobiológico en particular es el basado en poblaciones celulares [37]. Las variables de estado de estos modelos son las concentraciones de las principales células y factores bioquímicos involucrados en el modelo de remodelación ósea [37, 38].

Por ello estos modelos tienen la capacidad de ofrecer una representación detallada de los procesos biológicos involucrados en la remodelación ósea. Esto los convierte en herramientas adecuadas para analizar enfermedades que afectan a este proceso de remodelación como la osteoporosis postmenopausica [37, 39] así como también los tratamientos para combatirla [38, 40]. En este capítulo se describe uno de estos modelos mecanobiológicos basado en poblaciones celulares, en concreto el desarrollado por Martínez-Reina y cols [38], [40] y [41].

3.1 Ecuaciones diferenciales

El proceso de remodelación se puede describir como un balance celular utilizando las concentraciones celulares de: osteoblastos precursores OB_p , osteoblastos activos OB_a , osteoclastos precursores OC_p , osteoclastos activos OC_a y osteocitos O_t . Las concentraciones de células sin comprometer, OB_u y OC_u se han asumido constantes.

$$\frac{dOB_p}{dt} = D_{OB_u} \cdot \pi_{act,OB_u}^{TGF-\beta} \cdot OB_u + P_{OB_p} \cdot \Pi_{act,OB_p}^{Wnt} \cdot OB_p - D_{OB_p} \cdot \pi_{rep,OB_p}^{TGF-\beta} \cdot OB_p \quad (3.1)$$

$$\frac{dOB_a}{dt} = D_{OB_p} \cdot \pi_{rep,OB_p}^{TGF-\beta} \cdot OB_p - \Delta_{OB_a} \cdot OB_a \quad (3.2)$$

$$\frac{dOC_p}{dt} = D_{OC_u} \cdot \pi_{act,OC_u}^{RANKL} \cdot \pi_{act,OC_u}^{MSCF} \cdot OC_u - D_{OC_p} \cdot \pi_{act,OC_p}^{RANKL} \cdot OC_p \quad (3.3)$$

$$\frac{dOC_a}{dt} = D_{OC_p} \cdot \pi_{act,OC_p}^{RANKL} \cdot OC_p - A_{OC_a} \cdot \pi_{act,OC_p}^{TGF-\beta} \cdot OC_a \quad (3.4)$$

$$\frac{dO_t}{dt} = \eta \frac{df_{bm}}{dt} \quad (3.5)$$

Por otro lado, la evolución de la porosidad vascular y la fracción de volumen de la matriz ósea vienen dadas por:

$$\frac{\partial f_{vas}}{\partial t} = -\frac{\partial f_{bm}}{\partial t} = -k_{form} OB_a + k_{res} OC_a \quad (3.6)$$

La porosidad y la fracción volumétrica cambian al actuar los osteoclastos y los osteoblastos. A pesar de ello, como la capa de osteoide depositada por los osteoblastos mineraliza muy rápido en comparación con la remodelación, se supone que la matriz ósea esta mineralizada desde el principio. Además, se igualan las tasas de reabsorción de la matriz ósea y la de deposición de nueva matriz ósea [42].

Se muestra en la figura 3.1 una representación del modelo de poblaciones celulares descrito.

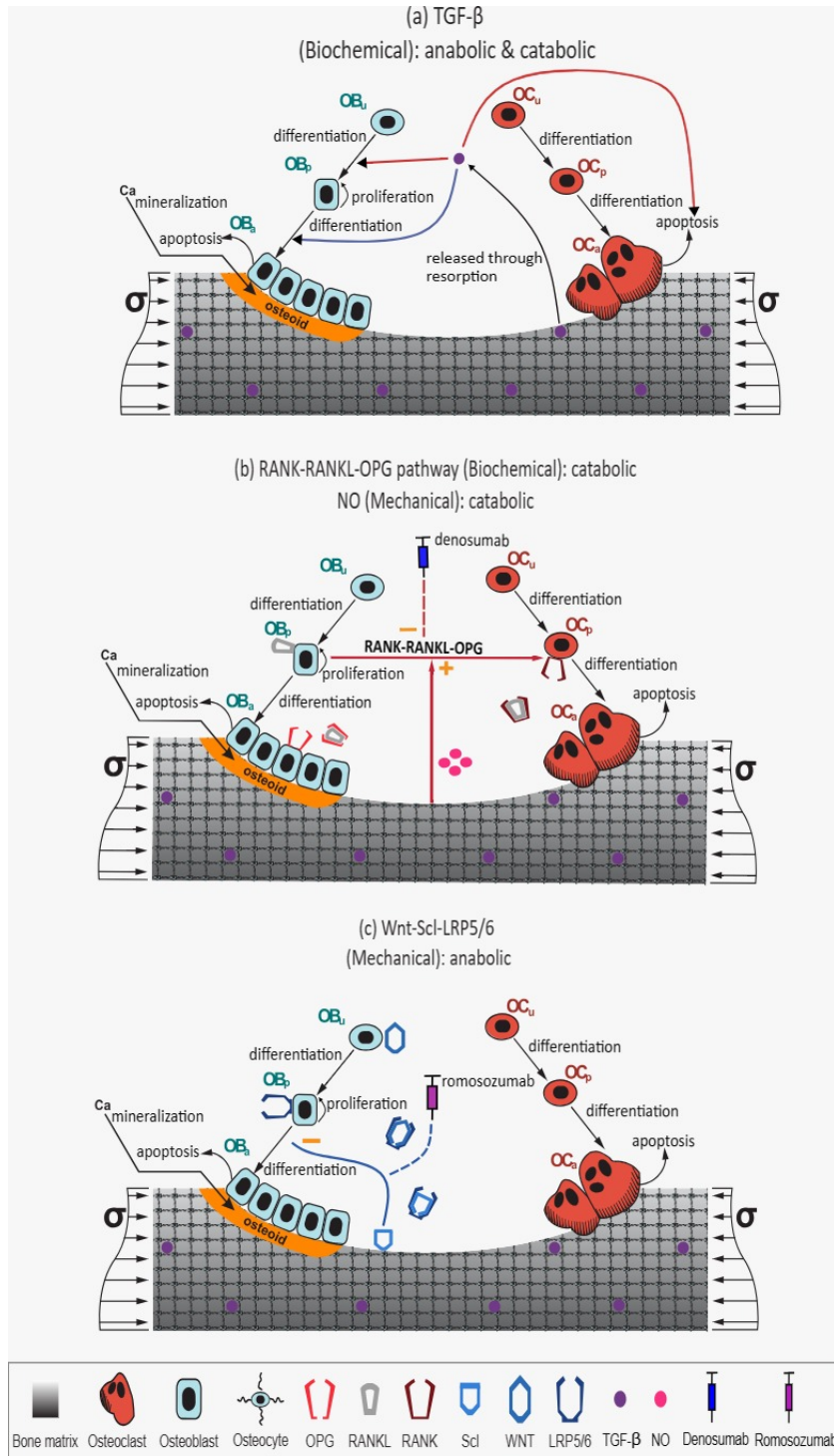


Figura 3.1 Representación esquemática del modelo de población de células óseas: se presentan las etapas de diferenciación de las células óseas junto con las interacciones bioquímicas y biomecánicas..

3.2 Parámetros que describen el modelo

En primer lugar, se van a exponer los parámetros que describen el modelo para su mayor comprensión a la hora de ver las ecuaciones. Se han recogido primero las cantidades dinámicas que gobiernan el problema [42] (Tabla 3.1) y a continuación, las constantes del modelo junto con una breve

descripción (tabla 3.3).

Tabla 3.1 Cantidades dinámicas en las ecuaciones de gobierno.

Símbolo	Descripción
OC_a	Densidad de osteoclastos activos, en pM
OB_p	Densidad de pre-osteoblastos, en pM
OB_a	Densidad de osteoblastos activos, en pM
OC_p	Densidad de osteoclastos precursores, en pM
O_t	Densidad de osteocitos, en pM
$TGF\beta$	Concentración de $TGF\beta$, en pM
RANK	Concentración de RANK, en pM
RANKL	Concentración de RANKL, en pM
OPG	Concentración de OPG, en pM
PTH	Concentración de PTH, en pM
f_{vas}	Porosidad vascular
f_{bm}	Porosidad fracción de volumen de matriz ósea
E	Módulo de Young, en MPa
d	Daño
ρ	Densidad aparente, en g/cm^3
v_m	Fracción de mineral en el hueso

Tabla 3.2 Constantes del modelo, sus valores, unidades y descripción.

Parámetro	Valor	Unidad	Descripción
D_{OB_u}	1.00E-02	día ⁻¹	Tasa de diferenciación de OB_u a OB_p
D_{OB_p}	7.05E-02	día ⁻¹	Tasa de diferenciación de OB_p a OB_a
A_{OB_a}	3.15E-01	día ⁻¹	Tasa de apoptosis de OB_a
D_{OB_a}	3.50E-02	día ⁻¹	Tasa de diferenciación de OB_a a O_t
D_{OC_u}	3.88E-02	día ⁻¹	Tasa de diferenciación de OC_u a OC_p
D_{OC_p}	1.21E-01	día ⁻¹	Tasa de diferenciación de OC_p a OC_a
A_{OC_a}	4.50E+00	día ⁻¹	Tasa de apoptosis de OC_a
OB_u	1.00E-02	pM	Concentración inicial de OB_u
OC_u	1.00E-02	pM	Concentración inicial de OC_u
η	4.14E-04	pM / %	Concentración de osteocitos en la matriz ósea
$K_{act}^{TGF\beta}$	5.63E-04	pM	Coficiente de activación de TGF relacionado con la diferenciación de OB_u a OB_p
$K_{rep}^{TGF\beta}$	1.75E-04	pM	Coficiente de represión de TGF relacionado con la diferenciación de OB_p a OB_a
$K_{[RANKL-PTH],act}$	6.50E-01	pM	Coficiente de activación para la producción de RANKL debida a la unión de PTH
$K_{[RANKL-PTH],rep}$	2.23E-01	pM	Coficiente de represión para la producción de RANKL debida a la unión de PTH
$K_{[RANKL-RANK],act}$	7.48E+00	pM	Coficiente de activación relacionado con la unión de RANKL y RANK
$N_{OC_p}^{RANK}$	4.16E+03	pM RANK / pM cells	Coficiente de represión relacionado con la unión de RANKL y RANK

K_{act}^{MCSF}	1.00E-03	pM	Coficiente de activación relacionado con la unión de $MCSF$ y OC_u
$P_{RANKL-d}$	00.00E+00	pM/día	Inyección externa de $RANKL$
$\alpha_{TGF\beta}$	1.00E+00	%	$TGF\beta$ almacenado en la matriz ósea
\tilde{D}_{PTH}	8.60E+01	día ⁻¹	Tasa de degradación de PTH
β_{PTH}	2.50E+02	pM PTH /día	Tasa de síntesis de PTH sistémico
β_{OPG}	1.30E+05	pM OPG /pM cell/día	Tasa mínima de producción de OPG por célula
\tilde{D}_{OPG}	3.50E-01	día ⁻¹	Tasa de degradación de OPG
$\tilde{D}_{OPG-RANKL}$	5.59E+02	día ⁻¹	Tasa de degradación de $OPG - RANKL$
$[OPG]_{max}$	1.31E+02	pM OPG	Máxima concentración posible de OPG
\tilde{D}_{RANKL}	4.00E-01	día ⁻¹	Tasa de degradación de $RANKL$
$\tilde{D}_{TGF\beta}$	2.00E+02	día ⁻¹	Tasa de degradación de $TGF\beta$
\tilde{D}_{Scl}	5.00E+00	día ⁻¹	Tasa de degradación de Scl
Scl_{max}	7.00E+01	pM [Scl]	Máxima cantidad de Scl
$\tilde{D}_{SclLRP5}$	5.00E+01	día ⁻¹	Tasa de degradación de $SclLRP5$
Wnt	1.70E+02	pM	Wnt
$XKAPPA$	7.00E-03	día ⁻¹	Tasa de mineralización
v_{max}	5.16E-01	$V_{mineral}/VRVE$	Volumen específico de mineral correspondiente al máximo contenido de Ca
ρ_m	1.410E+00	g/cm^3	Densidad de la fase orgánica
ρ_o	3.20E+00	g/cm^3	Densidad de mineral
N_f	1.00E+04	ciclos	Número de ciclos de fatiga
BW	6.00E+01	kg	Masa de la paciente
k_{res}	2.00E+03	% day ⁻¹ pM ⁻¹	Velocidad de reabsorción ósea
K_{form}	2.00E-01	% day ⁻¹ pM ⁻¹	Velocidad de formación ósea
ρ_m	3.20E+00	g/cm^3	Densidad de mineral
$N_{OC_p}^{RANK}$	1.00E+04	pM $RANK$ /pM cell	Número máximo de $RANK$ por OC_p
$D_{RANK-RANKL}$	1.01E+01	día ⁻¹	Tasa de degradación de $RANK - RANKL$
$k_D^{RANK-RANKL}$	1.00E+01	pM	Constante de disociación en la unión $RANK - RANKL$
$k_D^{OPG-RANKL}$	2.30E+03	pM	Constante de disociación en la unión $OPG - RANKL$
β_{Scl}	5.00E+04	pM Scl /pM cell día ⁻¹	Velocidad de producción de Scl
$\beta_{NO,Ot}$	1.39E+03	pM NO /pM cell día ⁻¹	Tasa de producción de NO por Ot
D_{NO}	2.10E-03	día ⁻¹	Tasa de aclaramiento de NO .
$[NO]_{max}$	2.00E+08	pM	Concentración máxima de NO
$k_{D,rep}^{NO}$	5.23E+02	pM	Constante de disociación de NO
$D_{Scl-SclAb}$	3.20E+00	día ⁻¹	Tasa de degradación de $Scl - SclAb$
$k_{Scl-SclAb}$	3.2E+00	pM	Constante de disociación de $Scl - SclAb$
$[RANKL]_{max}$	4.15E+01	pM	Concentración máxima de $[RANK]$
$k_{Wnt-LRP5/6}$	1.07E+03	pM	Constante de disociación de $Wnt - LRP5/6$
$k_{Scl-LRP5/6}$	8.57E+00	pM	Constante de disociación de $Scl - LRP5/6$

$MCSF$	1.00E-03	pM		Factor estimulante de colonias Macrófagos
λ_c	4.56E-01	-		Co-regulación de $[RANKL]$ a través de la concentración de PTH y NO
λ_s	7.46E+01	-		Co-regulación de $[RANKL]$ a través de la concentración de PTH y NO
β_{RANKL,OB_p}	2.14E+02.	pM KL/pM día ⁻¹	RAN- célula	Producción de $[RANKL]$ por OB_p
$\beta_{RANKL,Ot}$	2.83E+03	pM KL/pM día ⁻¹	RAN- célula	Producción de $[RANKL]$ por Ot
τ	1.20E-03	$\mu\epsilon$		Microdeformaciones en equilibrio
ω	9.50E-01	-		Valor de las funciones de regulación de la tensión homeostática
δ_{rep}	6.00E+00	MPa		Valor del estímulo que produce la respuesta semi-máxima
δ_{act}	1.19E+01	MPa		Valor del estímulo que produce la respuesta semi-máxima
$N_{OB_p}^{LRP5/6}$	5.00E+00	pM LRP5/6 / pM cell		Número de $LRP5/6$ por OB_p
ρ_{act}	0.00E+00	-		Respuesta mínima anticipada
α_{act}	1.00E+00	-		Respuesta máxima anticipada
ρ_{rep}	0.00E+00	-		Respuesta mínima anticipada
α_{rep}	1.00E+00	-		Respuesta máxima anticipada
k_1	1.79E-05	pM/días		Primera pendiente $\alpha_{TGF-\beta}$
k_2	1.05E-07	pM/días		Segunda pendiente $\alpha_{TGF-\beta}$
T_1	1.50E+01	años		Tiempo de cambio de la pendiente de α
k_3	1.12E-02	pM/días ²		Primera pendiente Slc
k_4	5.03E-04	pM/días ²		Segunda pendiente Slc
T_2	5.41E+01	años		Tiempo de cambio de la pendiente de Slc
k_5	1.00E-02	pM/días ²		Primera pendiente de P_RANKL_PMO
k_6	1.00E+00	pM/días ²		Segunda pendiente de P_RANKL_PMO
T_3	1.60E+02	días		Tiempo de cambio de la pendiente de P_RANKL_PMO
K_{m1}	5.61E+01	pM		Constante de Michaelis del denosumab
k_{el1}	1.00E+00	días ⁻¹		Ratio de eliminación
V_{c1}	7.79E+01	mL/kg		Volumen del compartimento central del denosumab
V_{max1}	3.65E+02	ng/kg/día		Velocidad máxima de reacción del denosumab
F_1	1.00E+00	-		Biodisponibilidad del denosumab
ζ_1	6.50E-01	-		Proporción de denosumab que se elimina del compartimento central y entra en el compartimento óseo
M_{den}	1.47E+02	kg/mol		Masa molecular denosumab
K_m	1.14e+04	pM		Constante de Michaelis del romosozumab
k_{el}	6.00E-02	días ⁻¹		Ratio de eliminación
V_c	8.08E+01	mL/kg		Volumen del compartimento central del romosozumab

V_{max}	2.88E-03	ng/kg/días	Velocidad máxima de reacción del romosozumab
ζ	5.00E-1	-	Proporción de lo que se elimina del compartimento central y entra en el compartimento óseo
k_a	4.50E-01	días ⁻¹	Coefficiente que modela la velocidad de absorción del tratamiento
M_{SclAb}	1.49E+02	kg/mol	Masa molecular del romosozumab

3.3 Porosidad en la matriz ósea

Una vez se han establecido los parámetros que componen el modelo, se observa que la representación de la microestructura ósea implica el uso de parámetros geométricos, siendo la porosidad vascular uno de los más relevantes.

$$f_{vas} = \frac{V_{vas}}{V_T} \quad (3.7)$$

Donde V_{vas} es el volumen de poros vasculares y V_T es el volumen de tejido.

$$f_{bm} = \frac{V_{bm}}{V_T} \quad (3.8)$$

Donde V_{bm} es el volumen de matriz ósea. De estas ecuaciones se deduce que:

$$f_{vas} + f_{bm} = 1 \quad (3.9)$$

3.4 Funciones de Hill

La función de Hill es una herramienta matemática ampliamente utilizada para describir el comportamiento de genes y moléculas en sistemas biológicos. Esta función se utiliza para modelar procesos biológicos en los que la respuesta de una molécula o un gen dependen de la concentración de un ligando o de la presencia de otros factores reguladores. La nomenclatura es la siguiente:

$$\pi_{act/rep,molculo}^{ligando} \quad (3.10)$$

donde, *ligando* es el factor bioquímico, *molécula* la célula ósea receptora y *activación/represión* dependiendo de si la unión del factor bioquímico y del receptor favorece o inhibe una determinada acción.

define matemáticamente como:

$$\pi_{act,A}^L = \frac{L}{L + K_{act,A}^L} \quad (3.11)$$

Donde: $\pi_{act,A}^L$ representa la fracción de receptor sobre A que se une a L para favorecer un determinado proceso celular de A .

$$\pi_{rep,A}^L = 1 - \pi_{act,A}^L = \frac{1}{1 + \frac{L}{K_{rep,A}^L}} \quad (3.12)$$

Donde: $\pi_{rep,A}^L$ representa la fracción de receptor sobre A que se une a L para reprimir un determinado proceso celular de A .

Donde: K_A^L es la constante de activación/represión de la unión del factor bioquímico y el receptor. Es posible representar matemáticamente los factores bioquímicos que afectan a la remodelación ósea utilizando las funciones de Hill.

Las funciones correspondientes al $TGF\beta$ son:

$$\pi_{act,OB_u/OC_a}^{TGF\beta} = \frac{TGF\beta}{TGF\beta + K_{OB_u/OC_a,act}^{TGF\beta}} \quad (3.13)$$

$$\pi_{rep,OB_p}^{TGF\beta} = \frac{TGF\beta}{1 + \frac{TGF\beta}{K_{OB_p,rep}^{TGF\beta}}} \quad (3.14)$$

donde

$$TGF\beta = \frac{\alpha_{TGF\beta} \cdot k_{res} \cdot O_{c_a}}{\tilde{D}_{TGF\beta}} \quad (3.15)$$

Las funciones correspondientes al RANKL son:

$$\pi_{act,OC_u/OC_p}^{RANKL} = \frac{[RANKL]}{[RANKL] + K_{OC_u/OC_p,act}^{RANKL}} \quad (3.16)$$

$$\pi_{act,OC_u}^{RANKL} = \pi_{act,OC_p}^{RANKL} \quad (3.17)$$

Donde el coeficiente de activación K es igual a $K_{A2[RANKL-RANK]}$.

Las funciones correspondientes al MCSF son:

$$\pi_{act,OC_p}^{MCSF} = \frac{MCSF}{MCSF + K_{OC_p}^{MCSF}} \quad (3.18)$$

Donde $K_{OC_p}^{MCSF} = K_{act}^{MCSF}$.

Las funciones correspondientes al PTH son:

$$\pi_{act,OB_a/OB_p}^{PTH} = \frac{PTH}{PTH + K_{OB_a/OB_p,act}^{PTH}} \quad (3.19)$$

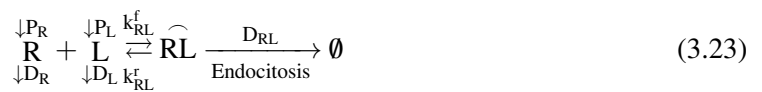
$$\pi_{act,OB_a}^{PTH} = \pi_{act,OB_p}^{PTH} \quad (3.20)$$

$$\pi_{rep,OB_a/OB_p}^{PTH} = \frac{PTH}{1 + \frac{PTH}{K_{OB_a/OB_p,act}^{PTH}}} \quad (3.21)$$

$$\pi_{rep,OB_a}^{PTH} = \pi_{rep,OB_p}^{PTH} \quad (3.22)$$

3.5 Modelo Receptor-Ligando

Se muestran a continuación las ecuaciones de 1 ligando con 1 receptor.



La R hace referencia a los receptores libre, L a los ligandos y \widehat{RL} los complejos formados por la unión de un receptor con un ligando.

Donde $k_{RL}^r, k_{RL}^f > 0$ son las tasas de unión de la reacción directa e inversa respectivamente y D_{RL} la tasa de degradación del complejo \widehat{RL} en el medio extracelular por medio de la endocitosis.

Suponiendo que las ecuaciones de reacción son de primer orden y la ley de acción de masas, las ecuaciones correspondientes a esas uniones de receptor-ligando son:

$$\frac{\partial \widehat{RL}}{\partial t} = k_{RL}^f \cdot R \cdot L - k_{RL}^r \cdot \widehat{RL} - D_{RL} \cdot \widehat{RL} \quad (3.24)$$

$$\frac{\partial R}{\partial t} = P_R - D_R \cdot R - k_{RL}^f \cdot R \cdot L + k_{RL}^r \cdot \widehat{RL} \quad (3.25)$$

$$\frac{\partial L}{\partial t} = P_L - D_L \cdot L - k_{RL}^f \cdot R \cdot L + k_{RL}^r \cdot \widehat{RL} \quad (3.26)$$

En este contexto, se asume que el equilibrio de las reacciones bioquímicas involucradas en la remodelación ósea se alcanza de manera rápida en comparación con los procesos celulares que las desencadenan. Por ello se aplica el concepto de estacionaridad, superíndice st , igualando a cero las derivadas temporales de las ecuaciones. De esta manera se obtiene:

$$k_{RL}^f \cdot R^{st} \cdot L^{st} - k_{RL}^r \cdot \widehat{RL}^{st} - D_{RL} \cdot \widehat{RL}^{st} = 0 \quad (3.27)$$

$$P_R - D_R \cdot R^{st} - k_{RL}^f \cdot R^{st} \cdot L^{st} + k_{RL}^r \cdot \widehat{RL}^{st} = 0 \quad (3.28)$$

$$P_L - D_L \cdot L^{st} - k_{RL}^f \cdot R^{st} \cdot L^{st} + k_{RL}^r \cdot \widehat{RL}^{st} = 0 \quad (3.29)$$

De la ecuación 3.27 se despeja \widehat{RL}^{st} :

$$\widehat{RL}^{st} = \frac{k_{RL}^f}{k_{RL}^r + D_{RL}} \cdot R^{st} \cdot L^{st} = \frac{1}{k_{RL}} \cdot R^{st} \cdot L^{st} \quad (3.30)$$

Donde

$$k_{RL} = \frac{k_{RL}^r + D_{RL}}{k_{RL}^f} \quad (3.31)$$

Sustituyendo esta igualdad en las ecuaciones 3.28 y 3.29 quedan:

$$P_R - D_R \cdot R^{st} + R^{st} \cdot L^{st} \cdot \left(\frac{k_{RL}^r}{k_{RL}} - k_{RL}^f \right) = 0 \quad (3.32)$$

$$P_L - D_L \cdot L^{st} + R^{st} \cdot L^{st} \cdot \left(\frac{k_{RL}^r}{k_{RL}} - k_{RL}^f \right) = 0 \quad (3.33)$$

Si se desarrolla la expresión $\frac{k_{RL}^r}{k_{RL}} - k_{RL}^f$, quedaría:

$$\frac{k_{RL}^r}{k_{RL}} - k_{RL}^f = k_{RL}^f \left(\frac{k_{RL}^r}{k_{RL}^r + D_{RL}} - 1 \right) = -\frac{D_{RL} \cdot k_{RL}^f}{k_{RL}^r + D_{RL}} = -\frac{D_{RL}}{k_{RL}} \quad (3.34)$$

Sustituyendo la igualdad en las ecuaciones 3.32 y 3.33:

$$D_R \cdot R^{st} + R^{st} \cdot L^{st} \cdot \frac{D_{RL}}{k_{RL}} = P_R \quad (3.35)$$

$$D_L \cdot L^{st} + R^{st} \cdot L^{st} \cdot \frac{D_{RL}}{k_{RL}} = P_L \quad (3.36)$$

Despejando R^{st} y L^{st} :

$$R^{st} = \frac{P_R}{D_R + \frac{D_{RL}}{k_{RL}} \cdot L^{st}} \quad (3.37)$$

$$L^{st} = \frac{P_L}{D_L + \frac{D_{RL}}{k_{RL}} \cdot R^{st}} \quad (3.38)$$

Sustituyendo la ecuación 3.37 en 3.38 se tiene:

$$L^{st} = \frac{P_L}{D_L + \frac{D_{RL}}{k_{RL}} \cdot \frac{P_R}{D_R + \frac{D_{RL}}{k_{RL}} \cdot L^{st}}} = \frac{P_L}{D_L + \frac{K_{RL} \cdot P_R}{D_{RL} \cdot D_R + L^{st}}} \quad (3.39)$$

Despejando P_L para que quede en función de L^{st} :

$$P_L = L^{st} \cdot D_L + \frac{P_R \cdot L^{st}}{\frac{K_{RL}}{D_{RL}} \cdot D_R + L^{st}} \quad (3.40)$$

$$D_L \cdot (L^{st})^2 + K_{RL} \cdot \frac{D_R \cdot D_L}{D_{RL}} \cdot L^{st} + P_R \cdot L^{st} = P_L \cdot L^{st} + P_L \cdot \frac{k_{RL}}{D_{RL}} \cdot D_R \quad (3.41)$$

$$D_L \cdot (L^{st})^2 + \left(K_{RL} \cdot \frac{D_R \cdot D_L}{D_{RL}} + P_R - P_L \right) \cdot L^{st} - P_L \cdot \frac{k_{RL}}{D_{RL}} \cdot D_R = 0 \quad (3.42)$$

De forma alternativa, el receptor se puede expresar de la misma forma:

$$D_R \cdot (R^{st})^2 + \left(K_{RL} \cdot \frac{D_R \cdot D_L}{D_{RL}} + P_L - P_R \right) \cdot R^{st} - P_R \cdot \frac{k_{RL}}{D_{RL}} \cdot D_L = 0 \quad (3.43)$$

3.5.1 Señal intracelular y respuesta celular

La cantidad de receptores ocupados determina la respuesta celular de la reacción $R + L$. Esto se representa a través de la fracción:

$$v_{RL} = \frac{\widehat{RL}}{R_{tot}} = \frac{\widehat{RL}}{R + \widehat{RL}} \quad (3.44)$$

Dado que se supone que las reacciones suceden tan rápidamente que se alcanza el equilibrio de reacción inmediatamente o, al menos, antes de que los receptores y sus ligandos influyan en la respuesta celular, se eliminó el superíndice "st".

$$\widehat{RL} = \frac{1}{k_{RL}} \cdot R \cdot L \quad (3.45)$$

$$v_{RL} = \frac{\frac{1}{k_{RL}} \cdot R \cdot L}{R + \frac{1}{k_{RL}} \cdot R \cdot L} = \frac{R \cdot L}{k_{RL} \cdot R + R \cdot L} = \frac{L}{L + k_{RL}} \quad (3.46)$$

En función de esta fracción se definen las funciones de activación y represión siguientes:

$$\pi_{act}^\alpha(v) = \frac{v}{\alpha + v \cdot (1 - \alpha)} \quad (3.47)$$

$$\pi_{rep}^\alpha(v) = 1 - \pi_{act}^\alpha(v) \quad (3.48)$$

Donde α es un parámetro de forma, al que normalmente se le da el valor $\alpha = 1$. Operando y sustituyendo las anteriores ecuaciones, se llega a:

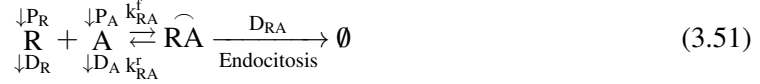
$$\pi_{act}^{\alpha}(v) = \frac{L}{L + \alpha \cdot k_{RL}} = \frac{L}{L + \kappa_{RL}} \quad (3.49)$$

$$\pi_{rep}^{\alpha}(v) = \frac{\alpha \cdot k_{RL}}{L + \alpha \cdot k_{RL}} = \frac{\kappa_{RL}}{L + \kappa_{RL}} \quad (3.50)$$

Donde κ_{RL} es el parámetro k_{RL} reescalado con el factor α .

3.5.2 Modelo receptor-ligando competitivo

Se toma la existencia un receptor (R) y dos ligandos para ese receptor (A y B).



Las ecuaciones diferenciales que modelan las reacciones de unión se exponen de la misma manera que si solo hubiera un receptor y un ligando:

$$\frac{\partial \widehat{\mathbf{RA}}}{\partial t} = k_{RA}^f \cdot \mathbf{R} \cdot \mathbf{A} - k_{RA}^r \cdot \widehat{\mathbf{RA}} - D_{RA} \cdot \widehat{\mathbf{RA}} \quad (3.53)$$

$$\frac{\partial \widehat{\mathbf{RB}}}{\partial t} = k_{RB}^f \cdot \mathbf{R} \cdot \mathbf{B} - k_{RB}^r \cdot \widehat{\mathbf{RB}} - D_{RB} \cdot \widehat{\mathbf{RB}} \quad (3.54)$$

$$\frac{\partial \mathbf{R}}{\partial t} = P_R - D_R \cdot \mathbf{R} - k_{RA}^f \cdot \mathbf{R} \cdot \mathbf{A} - k_{RB}^f \cdot \mathbf{R} \cdot \mathbf{B} + k_{RA}^r \cdot \widehat{\mathbf{RA}} + k_{RB}^r \cdot \widehat{\mathbf{RB}} \quad (3.55)$$

$$\frac{\partial \mathbf{A}}{\partial t} = P_A - D_A \cdot \mathbf{A} - k_{RA}^f \cdot \mathbf{R} \cdot \mathbf{A} + k_{RA}^r \cdot \widehat{\mathbf{RA}} \quad (3.56)$$

$$\frac{\partial \mathbf{B}}{\partial t} = P_B - D_B \cdot \mathbf{B} - k_{RB}^f \cdot \mathbf{R} \cdot \mathbf{B} + k_{RB}^r \cdot \widehat{\mathbf{RB}} \quad (3.57)$$

Se impone la estacionaridad:

$$\widehat{\mathbf{RA}}^{st} = \frac{1}{k_{RA}} \cdot A^{st} \cdot R^{st} \quad k_{RA} = \frac{k_{RA}^r + D_{RA}}{k_{RA}^f} \quad (3.58)$$

$$\widehat{\mathbf{RB}}^{st} = \frac{1}{k_{RB}} \cdot B^{st} \cdot R^{st} \quad k_{RB} = \frac{k_{RB}^r + D_{RB}}{k_{RB}^f} \quad (3.59)$$

De la misma forma se trata el resto de ecuaciones diferenciales, y sustituyendo las obtenidas anteriormente se tiene:

$$D_A \cdot A^{st} + R^{st} \cdot A^{st} \cdot \frac{D_{RA}}{k_{RA}} = P_A \quad (3.60)$$

$$D_B \cdot B^{st} + R^{st} \cdot B^{st} \cdot \frac{D_{RB}}{k_{RB}} = P_B \quad (3.61)$$

$$D_R \cdot R^{st} + R^{st} \cdot A^{st} \cdot \frac{D_{RA}}{k_{RA}} + R^{st} \cdot B^{st} \cdot \frac{D_{RB}}{k_{RB}} = P_R \quad (3.62)$$

Despejando los ligandos libres y el reactivo quedaría:

$$A^{st} = \frac{P_A}{D_A + \frac{D_{RA}}{k_{RA}} \cdot R^{st}} \quad (3.63)$$

$$B^{st} = \frac{P_B}{D_B + \frac{D_{RB}}{k_{RB}} \cdot R^{st}} \quad (3.64)$$

$$R^{st} = \frac{P_R}{D_R + \frac{D_{RA}}{k_{RA}} \cdot A^{st} + \frac{D_{RB}}{k_{RB}} \cdot B^{st}} \quad (3.65)$$

Sustituyendo las anteriores ecuaciones en la expresión 3.75 se obtiene una ecuación de tercer grado del tipo:

$$T_3 \cdot (R^{st})^3 + T_2 \cdot (R^{st})^2 + T_1 \cdot (R^{st}) + T_0 = 0 \quad (3.66)$$

Donde:

$$T_3 = D_R \cdot \frac{D_{RA}}{k_{RA}} \cdot \frac{D_{RB}}{k_{RB}} \quad (3.67)$$

$$T_2 = D_R \cdot \left(D_A \cdot \frac{D_{RB}}{k_{RB}} + D_B \cdot \frac{D_{RA}}{k_{RA}} \right) + (P_A + P_B - P_R) \cdot \frac{D_{RA}}{k_{RA}} \cdot \frac{D_{RB}}{k_{RB}} \quad (3.68)$$

$$T_1 = \frac{D_{RA}}{k_{RA}} \cdot D_B \cdot (P_A - P_R) + \frac{D_{RB}}{k_{RB}} \cdot D_A \cdot (P_B - P_R) + D_R \cdot D_A \cdot D_B \quad (3.69)$$

$$T_0 = -P_R \cdot D_A \cdot D_B \quad (3.70)$$

A continuación, se va a particularizar lo anteriormente visto para un receptor y ligandos específicos, es decir, el ligando "A" será a partir de ahora Scl (esclerostina), el ligando "B" Wnt y el receptor "R" LRP5/6.

De la ecuación 3.65 se obtenía sustituyendo R^{st} por LRP5/6:

$$[LRP5/6] = \frac{P_{LRP5/6}}{D_{LRP5/6} + \frac{D_{Wnt-LRP5/6}}{k_{Wnt-LRP5/6}} \cdot [Wnt] + \frac{D_{Scl-LRP5/6}}{k_{Scl-LRP5/6}} \cdot [Scl]} \quad (3.71)$$

Además

$$\begin{aligned} [LRP5/6]_{tot} &= [LRP5/6] + [LRP5/6 - Wnt] + [LRP5/6 - Scl] = \\ &= [LRP5/6] + \frac{[LRP5/6] \cdot [Wnt]}{k_{Wnt-LRP5/6}} + \frac{[LRP5/6] \cdot [Scl]}{k_{Scl-LRP5/6}} = \\ &= [LRP5/6] \cdot \left(1 + \frac{[Wnt]}{k_{Wnt-LRP5/6}} + \frac{[Scl]}{k_{Scl-LRP5/6}} \right) \end{aligned} \quad (3.72)$$

Para evitar resolver la ecuación cúbica 3.66, en Martin et al. [22] se simplifica manteniendo constante la concentración de proteínas Wnt en un valor de 170 pM. Esta suposición se fundamenta en la idea de que la degradación del complejo Wnt-LRP5/6 es insignificante, y que las células madre mesenquimales de la médula ósea (OB_u) son las principales productoras de Wnt, y esta población se considera constante en el modelo. En consecuencia, en este modelo se asume que la concentración de LRP5/6 no varía en función de $[Wnt]$ ni de $[Scl]$, sino que depende únicamente de OB_p .

$$[LRP5/6]_{tot} = N_{OB_p}^{LRP5/6} \cdot OB_p \quad (3.73)$$

Donde $N_{OB_p}^{LRP5/6}$ es el número de receptores de LRP5/6 por preosteoblasto.

Otra simplificación realizada en el artículo es que las reacciones de unión son muy rápidas frente a la evolución de las poblaciones celulares y la producción de ligandos que, la producción está equilibrada con la degradación, es decir:

$$Producción = Degradación \quad (3.74)$$

Generalizando la ecuación 3.75, se tiene que:

$$P_R = D_R \cdot R + \sum_I \frac{D_{RI}}{k_{RI}} \cdot R \cdot I \quad (3.75)$$

El término de producción es:

$$P_R = P_{R,b} + P_{R,d} \quad (3.76)$$

Donde $P_{R,b}$ es la producción en el cuerpo y $P_{R,d}$ una dosis externa.

$$P_{R,b} = \sum_{x,y} \beta_{R,x} \cdot \pi_{act,rep}^y \cdot x \cdot \left(1 - \frac{R}{R_{max}}\right) \quad (3.77)$$

Donde $\beta_{R,x}$ es la velocidad de producción de "R" por parte de "x", $\pi_{act,rep}^y$ una función de regulación por la especie "y", "x" es la concentración de células que producen "R", y " R_{max} " es la concentración de saturación.

Particularizando esta ecuación para la esclerostina (Scl):

$$P_{Scl,d} + P_{Scl,b} = D_{Scl} [Scl] + D_{Scl-LRP5/6} \cdot [Scl - LRP5/6] \quad (3.78)$$

$$P_{Scl,d} + \beta_{Scl} [Ot] \cdot \pi_{rep,Scl}^{\psi_{bm}} \cdot \left(1 - \frac{[Scl]}{Scl_{max}}\right) = D_{Scl} [Scl] + \frac{D_{Scl-LRP5/6} \cdot [Scl] \cdot [LRP5/6]}{k_{Scl-LRP5/6}} \quad (3.79)$$

A continuación, se sustituye en la anterior ecuación la expresión 3.72:

$$\begin{aligned} P_{Scl,d} + \beta_{Scl} [Ot] \cdot \pi_{rep,Scl}^{\psi_{bm}} \cdot \left(1 - \frac{[Scl]}{Scl_{max}}\right) &= \\ &= D_{Scl} [Scl] + \frac{D_{Scl-LRP5/6} \cdot [Scl] \cdot [LRP5/6]_{tot}}{k_{Scl-LRP5/6} \cdot \left(1 + \frac{[Wnt]}{k_{Wnt-LRP5/6}} + \frac{[Scl]}{k_{Scl-LRP5/6}}\right)} \end{aligned} \quad (3.80)$$

Esta ecuación se puede convertir en una de segundo grado del tipo:

$$A \cdot [Scl]^2 + B \cdot [Scl] + C = 0 \quad (3.81)$$

Siendo los coeficientes:

$$A = D_{Scl} + \frac{\beta_{Scl} [Ot] \cdot \pi_{rep,Scl}^{\psi_{bm}}}{[Scl]_{max}} \quad (3.82)$$

$$B = A \cdot k_{Scl-LRP5/6} \cdot \left(1 + \frac{[Wnt]}{k_{Wnt-LRP5/6}} \right) - \left(P_{Scl,d} + \beta_{Scl} [Ot] \cdot \pi_{rep,Scl}^{\psi_{bm}} \right) + D_{Scl-LRP5/6} [LRP5/6]_{tot} \quad (3.83)$$

$$C = -k_{Scl-LRP5/6} \cdot \left(1 + \frac{[Wnt]}{k_{Wnt-LRP5/6}} \right) \cdot \left(P_{Scl,d} + \beta_{Scl} [Ot] \cdot \pi_{rep,Scl}^{\psi_{bm}} \right) \quad (3.84)$$

El factor "A" es estrictamente positivo y "C" es estrictamente negativo. El producto de las dos raíces es $\frac{C}{A} < 0$. Además, existen dos soluciones reales, dado que $B^2 - 4AC > 0$ y, por lo tanto, una es positiva y la otra negativa. Sin embargo, solo tiene sentido físico la solución positiva, quedando:

$$[Scl] = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (3.85)$$

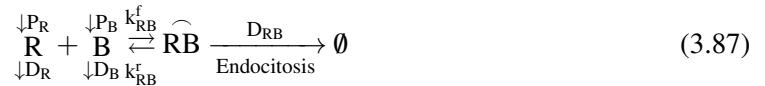
El procedimiento que se va a seguir partiendo de las ecuaciones detalladas anteriormente es el siguiente:

1. Se parte de la expresión 3.73: $[LRP5/6]_{tot} = N_{OB_p}^{LRP5/6} \cdot OB_p$.
2. Se sustituye en el coeficiente "B" de la expresión 3.83, con $[Ot]$ y $[Wnt]$ constantes. Además, teniendo en cuenta que $\pi_{rep,Scl}^{\psi_{bm}}$ y $P_{Scl,d}$ más las constantes son conocidas, se tienen todos los coeficientes de la ecuación de segundo grado 3.81.
3. De la expresión 3.85 se obtiene $[Scl]$.
4. Con la ecuación 3.71 se calcula $[LRP5/6]$.
5. Finalmente se calcularía $\pi_{act,OB_p}^{Wnt} = \frac{[Wnt] \cdot [LRP5/6]}{k_{Wnt-LRP5/6} \cdot [LRP5/6]_{tot}}$

3.5.3 Modelo receptor-ligando competitivo + anticuerpo

Este modelo se emplea para simular las interacciones en el complejo Scl-Wnt-LRP5/6 cuando se introduce un anticuerpo contra el Scl (Scl-Ab). Para mayor claridad, a partir de este punto, usaremos las letras "A" para representar al Scl, "B" para el Wnt, "R" para el LRP5/6 y "L" para denotar al Scl-Ab.

Al igual que en el caso anterior están las reacciones:



A las que se le añade la ecuación:



Estas reacciones de unión rigen las ecuaciones 3.53-3.57, excepto la 3.56. Además se le añade la expresión:

$$\frac{\partial \widehat{LA}}{\partial t} = k_{LA}^f \cdot L \cdot A - k_{LA}^r \cdot \widehat{LA} - D_{LA} \cdot \widehat{RA} \quad (3.89)$$

La ecuación 3.56 pasa a ser:

$$\frac{\partial A}{\partial t} = P_A - D_A \cdot A - k_{RA}^f \cdot R \cdot A + k_{RA}^r \cdot \widehat{RA} - k_{LA}^f \cdot L \cdot A + k_{LA}^r \cdot \widehat{LA} \quad (3.90)$$

De la expresión 3.89 se obtiene al igual que en los casos anteriores:

$$\widehat{LA}^{st} = \frac{1}{k_{LA}} \cdot A^{st} \cdot L^{st} \quad k_{LA} = \frac{k_{LA}^r + D_{LA}}{k_{LA}^f} \quad (3.91)$$

Imponiendo la estacionaridad:

$$L^{st} = \frac{P_L}{D_{LA} + \frac{D_{LA}}{k_{LA}} \cdot A^{st}} \quad (3.92)$$

$$B^{st} = \frac{P_B}{D_B + \frac{D_{RB}}{k_{RB}} \cdot R^{st}} \quad (3.93)$$

$$R^{st} = \frac{P_R}{D_R + \frac{D_{RA}}{k_{RA}} \cdot A^{st} + \frac{D_{RB}}{k_{RB}} \cdot B^{st}} \quad (3.94)$$

$$A^{st} = \frac{P_A}{D_A + \frac{D_{RA}}{k_{RA}} \cdot R^{st} + \frac{D_{LA}}{k_{LA}} \cdot L^{st}} \quad (3.95)$$

Se tienen cuatro ecuaciones con cuatro incógnitas que se simplificarán como se hizo anteriormente. De la última ecuación:

$$P_A = D_A \cdot A + D_{RA} \cdot \widehat{RA} + D_{LA} \cdot \widehat{LA} \quad (3.96)$$

Junto con:

$$R_{tot} = R + \widehat{RA} + \widehat{RB} = R \cdot \left(1 + \frac{A}{k_{AR}} + \frac{B}{k_{BR}} \right) \quad (3.97)$$

$$R = \frac{R_{tot}}{\left(1 + \frac{A}{k_{AR}} + \frac{B}{k_{BR}} \right)} \quad (3.98)$$

Y:

$$\widehat{LA} = \frac{L \cdot A}{k_{LA}} \quad (3.99)$$

Queda:

$$P_A = \left(D_A + \frac{D_{LA}}{k_{LA} \cdot L} \right) \cdot A + \frac{D_{RA}}{k_{RA}} \cdot \frac{A \cdot R_{tot}}{1 + \frac{A}{k_{AR}} + \frac{B}{k_{BR}}} \quad (3.100)$$

Ahora se suponen conocidos $B = [Wnt]$, $R_{tot} = [LRP5/6]_{tot}$ y $L = [Scl - Ab]$ conocido a través del modelo PK de romosozumab.

Y la ecuación será [41]:

$$\begin{aligned}
P_{Scl,d} + \beta_{Scl} [Ot] \cdot \pi_{rep,Scl}^{\psi_{bm}} \cdot \left(1 - \frac{[Scl]}{Scl_{max}}\right) &= \\
= D_{Scl} [Scl] + \frac{[Scl - Ab]}{k_{Scl-Scl-Ab}} \cdot D_{Scl-Scl-Ab} \cdot [Scl] + \frac{D_{Scl-LRP5/6} \cdot [Scl] \cdot [LRP5/6]_{tot}}{k_{Scl-LRP5/6} \cdot \left(1 + \frac{[Wnt]}{k_{Wnt-LRP5/6}} + \frac{[Scl]}{k_{Scl-LRP5/6}}\right)} & \quad (3.101)
\end{aligned}$$

3.5.4 Regulación del RANKL vía PTH y NO

Los factores PTH y NO influyen en el término $\pi_{act/rep,RANKL}^{PTH/NO}$ que aparece en el término de producción endógena de RANKL. Mientras que el PTH regula positivamente la producción de RANKL (fomentando la remodelación ósea), NO la regula negativamente. Esto se hace a través del estímulo mecánico que estimula a su vez la producción de NO por los osteocitos.

Dado que NO no interviene en ninguna reacción, la condición de equilibrio se establece por la clásica igualdad: *Producción = Degradación*.

$$P_{NO,b} + P_{NO,d} = D_{NO} \cdot [NO] \quad (3.102)$$

$$P_{NO,b} = \beta_{NO,Ot} [Ot] \cdot \pi_{act,NO}^{\psi_{bm}} \cdot \left(1 - \frac{[NO]}{[NO]_{max}}\right) \quad (3.103)$$

Que conducen a la expresión:

$$[NO] = \frac{P_{NO,d} + \beta_{NO,Ot} [Ot] \cdot \pi_{act,NO}^{\psi_{bm}}}{D_{NO} + \frac{\beta_{NO,Ot} [Ot] \cdot \pi_{act,NO}^{\psi_{bm}}}{[NO]_{max}}} \quad (3.104)$$

Para el PTH también se usa la expresión clásica, suponiendo:

1. PTH no está regulado por ningún factor, ni producido por ninguna célula, sino que lo produce la glándula paratiroidea a velocidad constante.
2. $PTH_{max} \rightarrow \infty$, es decir, que no se satura. Entonces $P_{PTH,b} = \beta_{PTH}$, por lo tanto:

$$[PTH] = \frac{\beta_{PTH} + P_{PTH,d}}{D_{PTH}} \quad (3.105)$$

Que resulta una constante si $P_{PTH,d} = 0$

Las funciones de regulación quedarían:

$$\pi_{act,RANKL}^{PTH} = \frac{[PTH]}{[PTH] + k_{D,act}^{PTH}} \quad (3.106)$$

$$\pi_{rep,RANKL}^{NO} = \frac{k_{D,rep}^{NO}}{[NO] + k_{D,rep}^{NO}} \quad (3.107)$$

Y finalmente la función corregulatoria:

$$\pi_{act/rep,RANKL}^{PTH,NO} = \lambda_s \cdot \left(\pi_{act,RANKL}^{PTH} + \pi_{rep,RANKL}^{NO}\right) + \lambda_c \cdot \pi_{act,RANKL}^{PTH} \cdot \pi_{rep,RANKL}^{NO} \quad (3.108)$$

Donde λ_s y λ_c son constantes que, respectivamente describen las influencias únicas y combinadas de las funciones respectivas activadora/represora.

La producción de NO está basada en el artículo de Martin et al. 2019 [22]), pero en lugar de tomar como estímulo mecánico la energía de deformación, se ha tomado la deformación principal máxima:

$$\pi_{act,NO}^{|\varepsilon|_{max}} = \rho_{act} + \frac{(\alpha_{act} - \rho_{act})|\varepsilon|_{max}}{\delta_{act}^{\gamma_{act}} + |\varepsilon|_{max}^{\gamma_{act}}} \quad (3.109)$$

El estímulo mecánico también influye en la producción de Scl:

$$\pi_{rep,Scl}^{|\varepsilon|_{max}} = \alpha_{rep} - \frac{(\alpha_{rep} - \rho_{rep})|\varepsilon|_{max}^{\gamma_{rep}}}{\delta_{rep}^{\gamma_{rep}} + |\varepsilon|_{max}^{\gamma_{rep}}} \quad (3.110)$$

Donde ρ , α , γ , δ son respectivamente la respuesta mínima anticipada, la respuesta máxima anticipada, el término sigmoideidad que influye en la inclinación de la respuesta y el valor del estímulo que produce la respuesta media máxima. Y $|\varepsilon|_{max}$ es la deformación principal máxima en valor absoluto.

Para los valores máximos y mínimos de estos factores se toma $\rho_{act} = \rho_{rep} = 0$ y $\alpha_{act} = \alpha_{rep} = 1$.

3.5.5 Via de señalización RANK-RANKL-OPG

Este apartado está basado en el artículo de Martin et al. 2019 [22]. En esta vía se utiliza la formulación del competitive binding pero con ligeros cambios, tanto en la definición de las constantes como en el uso de las ecuaciones.

RANK

En primer lugar se va a ver la formulación adoptada:

$$RANK = \frac{N_{OC_p}^{RANK} \cdot OC_p}{1 + \frac{[RANKL]}{k_D^{RANK-RANKL}}} \quad (3.111)$$

La formulación clásica es:

$$RANK = \frac{P_{RANK}}{D_{RANK} + \frac{D_{RANK-RANKL}}{k_D^{RANK-RANKL}} [RANKL]} = \frac{P_{RANK}/D_{RANK}}{1 + \frac{D_{RANK-RANKL}}{D_{RANK} \cdot k_D^{RANK-RANKL}} [RANKL]} \quad (3.112)$$

Ambas formulaciones son equivalentes si se redefinen las constantes. $\frac{P_{RANK}}{D_{RANK}} = N_{OC_p}^{RANK} \cdot OC_p$ y $\frac{D_{RANK-RANKL}}{D_{RANK} \cdot k_D^{RANK-RANKL}} = \frac{1}{k_D^{RANK-RANKL}}$, por lo que se toma:

$$D_{RANK} = 1 \quad D_{RANK-RANKL} = 1 \quad P_{RANK} = N_{OC_p}^{RANK} \cdot OC_p \quad (3.113)$$

OPG

Para OPG se utiliza la formulación clásica:

$$[OPG] = \frac{P_{OPG}}{D_{OPG} + \frac{D_{OPG-RANKL}}{k_D^{OPG-RANKL}} [RANKL]} \quad (3.114)$$

Con

$$P_{OPG} = \beta_{OPG} \cdot \pi_{rep,OB_a}^{PTH} \cdot OB_a \cdot \left(1 - \frac{OPG}{OPG_{max}}\right) \quad (3.115)$$

Que junto con la ecuación 3.114 y teniendo en cuenta que $P_{OPG,d} = 0$ conduce a:

$$[OPG] = \frac{\beta_{OPG} \cdot \pi_{rep,OB_a}^{PTH} \cdot OB_a}{D_{OPG} + \frac{D_{OPG-RANKL}}{k_D^{OPG-RANKL}} [RANKL] + \frac{\beta_{OPG} \cdot \pi_{rep,OB_a}^{PTH} \cdot OB_a}{OPG_{max}}} \quad (3.116)$$

Esta ecuación se asemeja a la clásica 3.93 haciendo que:

$$P_B = \beta_{OPG} \cdot \pi_{rep,OB_a}^{PTH} \cdot OB_a \quad D_B = D_{OPG} + \frac{P_B}{OPG_{max}} \quad (3.117)$$

RANKL

Se utiliza la vía clásica para el receptor (ecuación 3.94):

$$[RANKL] = \frac{P_{RANKL}}{D_{RANKL} + \frac{D_{RANK-RANKL}}{k_D^{RANK-RANKL}} [RANK] + \frac{D_{OPG-RANKL}}{k_D^{OPG-RANKL}} [OPG]} \quad (3.118)$$

La definición de P_{RANKL} quedaría:

$$P_{RANKL,b} = \beta_{RANKL,Ot} \cdot [Ot] \cdot \left(1 - \frac{[RANKL]_{tot}}{[RANK]_{max}}\right) + \beta_{RANKL,OB_p} \cdot \pi_{act/rep,RANKL}^{PTH,NO} \cdot OB_p \cdot \left(1 - \frac{[RANKL]_{tot}}{[RANK]_{max}}\right) \quad (3.119)$$

Con

$$[RANKL]_{tot} = [RANKL] \cdot \left(1 + \frac{[RANK]}{k_D^{RANK-RANKL}} + \frac{[OPG]}{k_D^{OPG-RANKL}}\right) \quad (3.120)$$

Las constantes:

$$x_1 = \beta_{RANKL,Ot} \cdot [Ot] \quad (3.121)$$

$$x_2 = \beta_{RANKL,OB_p} \cdot \pi_{act/rep,RANKL}^{PTH,NO} \cdot OB_p \quad (3.122)$$

Se suponen conocidas en Martin et al. 2019, con los valores de $[RANKL]$, $[RANK]$ y $[OPG]$ obtenidos en la iteración anterior. En este caso P_{RANKL} sería conocido.

Si en la definición de $P_{RANKL,b}$ se usara $[RANKL]$ en lugar de $[RANKL]_{tot}$ que ha introducido, se reordenaría la ecuación de la siguiente manera:

$$[RANKL] = \frac{P_{RANKL,d} + x_1 + x_2}{D_{RANKL} + \frac{x_1+x_2}{[RANKL]_{max}} + \frac{D_{RANK-RANKL}}{k_D^{RANK-RANKL}} \cdot [RANK] + \frac{D_{OPG-RANKL}}{k_D^{OPG-RANKL}} \cdot [OPG]} \quad (3.123)$$

Que coincide con la ecuación 3.94 haciendo:

$$P_R = P_{RANKL,d} + \beta_{RANKL,Ot} \cdot [Ot] + \beta_{RANKL,OB_p} \cdot \pi_{act/rep,RANKL}^{PTH,NO} \cdot OB_p \quad (3.124)$$

$$D_R = D_{RANKL} + \frac{\beta_{RANKL,Ot} \cdot [Ot] + \beta_{RANKL,OB_p} \cdot \pi_{act/rep,RANKL}^{PTH,NO} \cdot OB_p}{[RANKL]_{max}} \quad (3.125)$$

3.5.6 Forma de uso de las ecuaciones obtenidas

La forma de implementar las ecuaciones de Martin es parecida a una algoritmo de integración paso a paso, pero en el caso de una ecuación algebraica.

1. En primer lugar se utilizan las ecuaciones 3.111 y 3.116 con el valor de [RANKL] obtenido en el paso anterior para calcular [RANK] y [OPG].
2. A continuación, con los valores obtenidos se utiliza 3.120 para calcular $[RANKL]_{tot}$ y con este junto con la ecuación 3.119 se calcula P_{RANKL} .
3. Finalmente se usa la ecuación 3.123 para introducirla en el integrador de ecuaciones diferenciales algebraicas (en Matlab se usaría la función ode23t)

Una alternativa a esto sería utilizar la solución de la ecuación de tercer grado 3.66 con la redefinición de las constantes 3.113, 3.117 y 3.124.

En cualquier caso, ello implica también usar valores de la iteración anterior, en las poblaciones celulares implicadas y en las funciones de regulación (π).

Si se mantiene el uso de $[RANKL]_{tot}$ en 3.119, se podría calcular en función de los valores de [RANKL], [RANK] y [OPG] del paso anterior, ya que es de esperar que $\frac{[RANKL]_{tot}}{[RANKL]_{max}}$ sea pequeño frente a 1 y, entonces las variaciones de $[RANKL]_{tot}$ tendrían poca importancia.

En este caso P_{RANKL} se seguiría calculando con 3.119 (añadiendo $P_{RANKL,d}$) y se seguiría usando la redefinición de las variables 3.113 y 3.117.

3.6 Inclusión del daño

La acumulación de microfisuras en un volumen particular de material se aborda en el modelo utilizando un enfoque de Mecánica de Daño Continuo [43]. El daño microestructural es una variable que está relacionada con la densidad de microfisuras en un volumen de material y la pérdida de rigidez. Esta variable está definida de manera que $d \in [0, 1]$, donde $d = 0$ corresponde a un estado no dañado y $d = 1$ a una situación de fractura o fallo local. A medida que los osteoclastos eliminan el tejido óseo dañado, mientras que los osteoblastos depositan osteoide inicialmente intacto, se considera un equilibrio de microdaño entre la eliminación en la remodelación ósea (\dot{d}_R) y la acumulación debida a la carga por fatiga (\dot{d}_A). La ley de evolución del daño puede expresarse como:"

$$\dot{d} = \dot{d}_A - \dot{d}_R \quad (3.126)$$

Donde \dot{d}_A es la tasa de acumulación de daño debido a la carga por fatiga y \dot{d}_R es la tasa de eliminación de daño por la remodelación ósea. Esta última se evalúa asumiendo que el daño está uniformemente distribuido en todo el elemento de volumen representativo. Por lo tanto, la cantidad de daño reparado es proporcional al daño presente en ese volumen y al volumen de tejido que se está reabsorbiendo, \dot{V}_r , a través de la fracción que este volumen representa dentro del volumen de la matriz ósea:

$$\dot{d}_R = d \frac{\dot{V}_r}{V_{bm}} = d \frac{k_{res} \cdot Oc_a}{f_{bm}} \quad (3.127)$$

La acumulación de daño se evalúa siguiendo el procedimiento descrito en [44, 35]. Este procedimiento se basa en los resultados de los ensayos experimentales de fatiga realizados por Pattin et al. [45], quienes proporcionaron la evolución del daño en función del nivel de deformación y el número de ciclos. Esta evolución fue modelada matemáticamente por García-Aznar [46] para obtener la siguiente ecuación diferencial bajo tensiones de tracción:

$$\dot{d}_A = \dot{N} \frac{C_1}{C_2 \gamma_f} (1-d)^{1-\gamma_f} \epsilon_{max} e^{-C_2(1-d)^{\gamma_f}} \quad (3.128)$$

Donde \dot{N} es el número de ciclos aplicados por unidad de tiempo y ϵ_{max} es la máxima deformación principal expresada en microestrías. El resto de los parámetros y constantes del modelo son:

$$C_1 = \frac{e^{C_2} - 1}{K_f([Ca])}; \quad \delta_f = 14.1; \quad (3.129)$$

$$\gamma_f = 0.018(\varepsilon_{max} - 4100) + 12; \quad C_2 = -20; \quad (3.130)$$

Donde $K_f([Ca])$ es una función del contenido mineral que será definida a continuación. Los ensayos experimentales realizados por Pattin et al. [45] incluyeron una estimación de la vida a fatiga, N_f , que se relacionó con la deformación mediante la siguiente expresión:

$$N_f = \frac{K_f}{\varepsilon_{max}^{\delta_f}} \quad (3.131)$$

Donde se asumió que K_f era constante y igual a 1.445×10^{53} en tracción. Martínez-Reina et al. [44] introdujo una corrección en K_f para considerar la degradación de las propiedades de fatiga con el aumento del contenido mineral. Se asignó una vida de $N_f = 10^7$ ciclos al límite de fatiga, que generalmente se asume que ocurre para una fracción dada de la deformación última en tracción, ε_u/β , donde el parámetro β depende del tipo de material [47] y se asumió $\beta = 2$ para el hueso [44] con buenos resultados. Entonces, K_f se obtuvo a partir de la Ec. (3.131) como:

$$K_f([Ca]) = 10^7 \left(\frac{\varepsilon_u([Ca])}{\beta} \right)^{\delta_f} \quad (3.132)$$

Donde la deformación última en tracción depende de la concentración de calcio en la matriz ósea, [Ca], como mostró Currey [48]. En Martínez-Reina et al. [44], se ajustó la siguiente regresión a partir de los resultados experimentales presentados por Currey [48]:

$$\log \varepsilon_u = 31.452 - 11.341 \log[Ca] \quad (3.133)$$

Donde ε_u se expresa en microestrías y la concentración [Ca] se expresa en miligramos de calcio por gramo de matriz ósea. Esta concentración está directamente relacionada con la fracción de ceniza, α . Más precisamente, se asumió la relación $[Ca] = 398.8 \alpha$, basada en los pesos moleculares de la hidroxiapatita y el colágeno tipo I [44].

3.7 Inclusión de la mineralización

Tal como se menciona previamente (ver sección 2.2), los huesos están formados por tres componentes principales: una fase mineral, una fase orgánica y una cantidad de agua. Por lo tanto, al considerar un volumen de referencia, denotado como V_{RVE} y equivalente al volumen total V_T que incluye también los poros dentro del material óseo, podemos establecer lo siguiente:

$$V_T = V_{bm} + V_{vas} = V_m + V_o + V_w + V_{vas} \quad (3.134)$$

Los osteoblastos, durante el proceso de formación del hueso, depositan osteoide. Esta sustancia inicialmente consiste principalmente en la fase orgánica y agua. Sin embargo, a medida que ocurre la mineralización, parte de esta agua se convierte en la fase inorgánica, es decir, en minerales. Este proceso de transformación se divide en tres fases [37]:

- **Fase Inicial o Tiempo de Retraso de la Mineralización:** En esta etapa inicial, no se produce la mineralización de manera significativa. Por esta razón, también se le llama "tiempo de retraso de la mineralización". Esta fase puede durar de 6 a 22 días.

- **Fase Primaria:** La fase primaria es muy rápida y ocurre en cuestión de días. Durante esta fase, se alcanza alrededor del 70% de la mineralización máxima.
- **Fase Secundaria:** La fase secundaria se caracteriza por una tasa de mineralización más lenta. Aquí, los minerales se agregan al tejido óseo a una velocidad que disminuye exponencialmente a medida que el tejido se satura con minerales. Esta fase puede extenderse desde 6 meses hasta varios años, dependiendo de diversos factores.

El contenido de mineral en el hueso, que normalmente se mide mediante la fracción de mineral, α : la relación entre la masa de mineral, m_m , y la masa seca, que es la suma de las fases orgánica y mineral [37].

$$\alpha = \frac{m_m}{m_o + m_m} = \frac{\rho_m V_m}{\rho_o V_o + \rho_m V_m} \quad (3.135)$$

Dando esa misma ecuación pero en volúmenes específicos, dividiendo entre V_b :

$$\alpha = \frac{\rho_m v_m}{\rho_o v_o + \rho_m v_m} \quad (3.136)$$

La densidad tisular vendrá dada por:

$$\rho_t = \frac{m_w + m_m + m_o}{V_{bm}} = \rho_w v_w + \rho_m v_m + \rho_o v_o \quad (3.137)$$

Donde $\rho_w = 1g/cm^3$

Manteniéndose que:

$$v_o + v_m + v_w = 1 \quad (3.138)$$

$$\dot{v}_m = \frac{\dot{V}_m}{V_{bm}} - v_m \frac{\dot{V}_{bm}}{V_{bm}} = \dot{v}_m \Big|_{v_{bm}=cte} - v_m \frac{\dot{V}_{bm}}{V_{bm}} \quad (3.139)$$

El primer término se refiere a los cambios en el contenido mineral debido a la mineralización o reabsorción ósea. Es como si consideráramos la variación en el contenido mineral si la cantidad de tejido óseo (representada por V_{bm}) permaneciera constante.

El segundo término está relacionado con la variación en la porosidad. Para comprender esto, es importante destacar que el contenido de minerales disminuye cuando se deposita osteoide. Esto se debe a que el osteoide no contiene minerales y, en cambio, contribuye a aumentar el volumen de la matriz ósea. Como resultado, esto reduce la concentración de minerales en el tejido.

Ambos términos serán analizados por separado:

$$\dot{v}_m \Big|_{v_{bm}=cte} = \dot{v}_m \Big|_{mineralisation} - \dot{v}_m \Big|_{resorption} \quad (3.140)$$

Se asume que v_m que está gobernado por un modelo saturado:

$$\dot{v}_m \Big|_{mineralization} = K (v_m^{max} - v_m) \quad (3.141)$$

Esto conduce a una solución exponencial, que se asemeja bastante bien a la respuesta general del modelo anterior. Para un valor específico de la constante $K = 0.007$, se observa una rápida tasa de mineralización inicial (correspondiente a la fase primaria), que luego se desacelera a medida que transcurre más tiempo (fase secundaria). Además, se estableció un valor máximo para el contenido mineral, $v_m^{max} = 0.516$, de manera que, junto con el valor inicial $v_o = 3/7$ previamente mencionado, esto resulta en la obtención de la densidad tisular máxima, $\rho_t^{max} = 2.31g/cm^3$ [49]. La constante K se define fija en este trabajo.

Es importante tener en cuenta que el mineral es disuelto por los osteoclastos y, por lo tanto, se elimina de la matriz ósea. Por lo tanto, la cantidad de mineral eliminada debido a la reabsorción es directamente proporcional al contenido mineral presente en el tejido, v_m , y a la proporción de tejido que está siendo reabsorbido dentro de la matriz ósea, representada como $\frac{\dot{V}_r}{V_{bm}}$:

$$\dot{v}_m)_{resorption} = v_m \frac{\dot{V}_r}{V_{bm}} = v_m \frac{\dot{V}_r/V_{RVE}}{V_{bm}/V_{RVE}} = v_m \frac{\dot{V}_r/V_{RVE}}{f_{bm}} = \frac{v_m}{f_{bm}} k_{res} O_{Ca} \quad (3.142)$$

$$\dot{v}_m = K (v_m^{max} - v_m) - \frac{v_m}{f_{bm}} k_{res} O_{Ca} - v_m \frac{\dot{f}_{bm}}{f_{bm}} \quad (3.143)$$

Donde se ha usado que $\frac{\dot{V}_{bm}}{V_{bm}} = \frac{\dot{f}_{bm}}{f_{bm}}$. Teniendo en cuenta el balance entre la formación y la reabsorción:

$$\dot{v}_m = K (v_m^{max} - v_m) - \frac{v_m}{f_{bm}} k_{form} O_{Ca} \quad (3.144)$$

Finalmente la densidad aparente:

$$\rho = \frac{m_w + m_m + m_o}{V_{RVE}} = \frac{m_w + m_m + m_o}{V_{bm}} \frac{V_{bm}}{V_{RVE}} = \rho_t f_{bm} \quad (3.145)$$

3.7.1 Degradación de las propiedades de la fatiga por mineralización

A mayor fase mineral, más rígido es el hueso, a pesar de que aumenta la fragilidad. A mayor contenido mineral, mayor degradación de las propiedades de fatiga. Se consideran las siguientes hipótesis:

- La forma de la curva d-N se mantiene independiente del contenido de mineral.
- Solo la vida de fatiga se ve afectada por el contenido de mineral, al cambiar K_t , manteniendo constante el exponente δ_t .
- El límite de fatiga es 10^7 ciclos. Se toma como valor más común $\beta = 2$.

K_t es obtenido mediante:

$$K_t([Ca]) = 10^7 \left(\frac{\epsilon_u([Ca])}{\beta} \right)^{\delta_t} \quad (3.146)$$

ϵ_u depende de la concentración de [Ca] y se expresa de la siguiente forma [?]:

$$\log \epsilon_u = 25.452 - 11.341 \log [Ca] \quad (3.147)$$

3.8 Osteoporosis (PMO)

Se presenta a continuación la inclusión de la enfermedad de la osteoporosis postmenopáusica (PMO) en el modelo [50].

Tanto la edad como la menopausia pueden producir osteoporosis, en base a ellos se va a modelar la PMO en este modelo. Se simula de dos maneras el efecto de la edad en el hueso: decremento en la concentración de TGF- β y incremento en la producción de esclerostina. Para tener en cuenta el efecto de la menopausia se considera un incremento de la producción de RANKL.

3.8.1 Inclusión de la edad: Disminución de la liberación de TGF- β de la matriz ósea por reabsorción

El efecto de la edad se modela en base a los resultados de Nicolas et al. [51] y de Lemaire et al [52], es decir reduciendo el contenido de TGF- β en la matriz ósea. En este modelo a diferencia

que en Lemaire et al. quienes modelaron una disminución abrupta de la concentración de TGF- β , ahora, el TGF- β disminuye continuamente con el tiempo. Dado que Riggs et al. [53] informaron de disminución en la masa ósea del hueso trabecular vertebral a partir de la edad $t_{mat} = 20$ años, edad a la que comienza a disminuir la masa ósea, se ha asumido que los efectos comienzan a esa edad. Nicolas et al.[51] concluyeron que hay una disminución de con el tiempo; sin embargo, cuando compararon dos grupos de edad (20-29 años y 50-59 años), no pudieron encontrar diferencias significativas. Otros autores han encontrado resultados más concluyentes que confirmarían la correlación negativa entre la edad y el contenido de TGF- β tanto en suero [54] como en hueso [55]. En este último caso, hay una clara reducción del contenido de TGF- β dentro del intervalo entre los 30 y 50 años, y no hay diferencia después de los 50 años de edad. Basándonos en estos hallazgos, hemos utilizado una función bilineal para simular la disminución de TGF- β :

$$\alpha_{TGF-\beta}(t) = \begin{cases} \alpha_{TGF-\beta,0} & \text{para } TSSM < 0 \\ \alpha_{TGF-\beta,0} - k_1 TSSM & \text{para } 0 \leq TSSM < T_1 \\ \alpha_{TGF-\beta,0} - k_1 T_1 - k_2 (TSSM - T_1) & \text{para } T_1 \leq TSSM \end{cases} \quad (3.148)$$

Donde $TDME = t - t_{mat}$ se expresa en días, y t_{mat} es la edad en la que comienza a disminuir la masa ósea. Riggs et al. [53] midieron el descenso a partir de los 20 años, la edad de madurez esquelética. A esa diferencia se ha llamado TDME, en referencia a "Tiempo desde la Madurez Esquelética". Las pendientes k_1 y k_2 miden la tasa de disminución de la concentración de TGF- β en la matriz ósea con el tiempo. Se ha asumido que esta tasa de disminución puede variar después de que hayan transcurrido T_1 días desde t_{mat} . T_1 se ajusta junto con k_1 y k_2 para que se ajuste a la disminución observada de la masa ósea con la edad. Es importante señalar que hemos impuesto una restricción en las constantes k_1, k_2, T_1 tal que $\alpha_{TGF-\beta}(95 \text{ años}) \geq 0.9$. Esto se hizo para evitar la pérdida total de hueso que valores más bajos de $\alpha_{TGF-\beta}$, sistemáticamente producían en las simulaciones de ARO+PMO. Esta restricción también está justificada por un trabajo reciente [56], en el cual se demuestra que TGF- β desempeña un papel fundamental en la coordinación de la secuencia reabsorción-reversión-formación de la BMU y, por lo tanto, su valor no puede ser muy pequeño. Dado que la liberación de TGF- β de la matriz ósea a través de la reabsorción aumenta la apoptosis de los osteoclastos, una disminución en esa liberación puede aumentar la actividad osteoclástica que finalmente conduce a la pérdida ósea.

3.8.2 Inclusión de la edad: Incremento en la producción de esclerostina

Se asumió en [22] que el aumento en la concentración sérica de esclerostina era un efecto de la menopausia. Sin embargo, varios estudios [57, 58] sugirieron que esto debe ser un efecto de la edad, ya que los niveles crecientes de esclerostina se observan en mujeres premenopáusicas [57] y también en hombres [58]. En base a estos estudios, se ha considerado un aumento en la producción externa de esclerostina, $P_{Scl,d}$.

A pesar de que en Roforth et al. [59] se sugiere que el envejecimiento podría estar relacionado con una disminución en la eliminación de esclerostina de la circulación, pero también con un aumento en la producción de esclerostina por fuentes no relacionadas con el sistema óseo. De hecho, la esclerostina circulante puede derivarse no solo de los osteocitos [60], sino también de varias otras fuentes, como células mieloides [61, 62], células musculares lisas vasculares [63] y el riñón [60]. En este modelo se ha optado por una producción externa de esclerostina y no se ha considerado la reducción a través de otro mecanismos \tilde{D}_{Scl} :

$$P_{Scl,d}(t) = \begin{cases} 0 & \text{para } TSSM < 0 \\ k_3 TSSM & \text{para } 0 \leq TSSM < T_2 \\ k_3 T_2 + k_4 (TSSM - T_2) & \text{para } T_2 \leq TSSM \end{cases} \quad (3.149)$$

Donde las pendientes k_3 y k_4 miden la tasa de aumento en la producción externa de esclerostina. Se asume que la tasa de producción externa de esclerostina puede variar después de que hayan transcurrido T_2 días desde t_{mat} . La esclerostina inhibe la proliferación de OB_p y eventualmente conduce a una población menor de OB_a , lo que reduce la tasa de formación de nuevo tejido en eventos de remodelación y, en consecuencia, disminuye la densidad mineral ósea (BMD) si sus niveles aumentan.

3.8.3 Inclusión de la Menopausia: Incremento de la producción de RANKL

Se va a considerar un producción externa de RANKL, $P_{\text{RANKL}}^{\text{ED}}$, debida a la deficiencia de estrógenos como en los estudios [64, 39]. Se asume que dicha producción aumenta con el tiempo tras la menopausia siguiendo una función bilineal definida por los parámetros k_5, k_6 y T_3 :

$$P_{\text{RANKL}}^{\text{ED}}(t) = \begin{cases} 0 & \text{para } TSM < 0 \\ k_5 TSM & \text{para } 0 \leq TSM < T_3 \\ k_5 T_3 + k_6 (TSM - T_3) & \text{para } T_3 \leq TSM \end{cases} \quad (3.150)$$

Donde $T_{\text{SM}} = t - t_{\text{meno}}$ es el tiempo transcurrido desde la menopausia y se expresa en días. Siguiendo a Nordin et al. [65], hemos asumido aquí que $t_{\text{meno}} = 49$ años, siendo esta la edad a la que aparece la menopausia. Esta producción externa de RANKL estaría en concordancia con las conclusiones de Eghbali-Fatourehchi et al. [66] sobre el aumento de la expresión de RANKL por parte de las células estromales y los linfocitos con una disminución de la producción de estrógenos. Esta producción aumentaría los niveles de RANKL y promovería la diferenciación de O_{ca} , desequilibrando los eventos de remodelación ósea a favor de la resorción y, por lo tanto, conduciendo a la pérdida ósea.

3.8.4 Modelos PK-PD de Denosumab y Romosozumab

Los estudios in silico (Simulaciones computacionales) de tratamientos farmacológicos con modelos de remodelación ósea hacen uso de modelos farmacocinéticos (PK) y farmacodinámicos (PD). Los modelos PK simulan la absorción, distribución y eliminación del fármaco en el cuerpo y se desarrollan en conjunto con los modelos PD, que analizan el efecto del fármaco en el cuerpo. La combinación de ambos se denomina generalmente modelo PK-PD y su desarrollo es esencial para una simulación precisa del efecto del fármaco en la respuesta ósea.

En un trabajo previo [38], se propuso un nuevo modelo PK-PD para Denosumab (Dmab), que agregó el compartimento óseo a un modelo PK de un solo compartimento ya existente [67] (en un recuadro punteado en la figura 3.2) para simular cómo el fármaco se distribuye en el hueso y cómo interactúa con RANKL para inhibir la unión de RANK-RANKL. Basado en el modelo desarrollado por [37, 68], en el cual solo se incluyó el compartimento central, se asume que solo una fracción de lo que se elimina ($1 - \zeta_1$) se excreta en la orina, mientras que el resto (ζ_1) es el flujo de Dmab al compartimento óseo (ver figura 3.2). Esta última fracción se denomina $P_{\text{Dmab}_{BC}}$, es decir, el término de producción de Dmab en el compartimento óseo durante la unión competitiva entre RANKL, RANK y OPG:

$$P_{\text{Dmab}_{BC}} = \zeta_1 \left[k_{el1} [Dmab]_{CC} + \frac{V_{max1}}{V_{c1}/F_1} \frac{[Dmab]_{CC}}{K_{m1} + [Dmab]_{CC}} \right] \quad (3.151)$$

Donde el término entre corchetes en la Ec. 3.151 representa la eliminación del compartimento central en el modelo de [67]. Todas las constantes relacionadas con la PK de Dmab se han extraído del modelo de [37].

Se sigue el mismo procedimiento para modificar un modelo PK-PD existente de Romosozumab (Rmab) [41] (en un recuadro punteado en la figura 3.3) al agregar el compartimento óseo. En el modelo propuesto por Martin et al. [41], fue la concentración de Rmab en el compartimento central

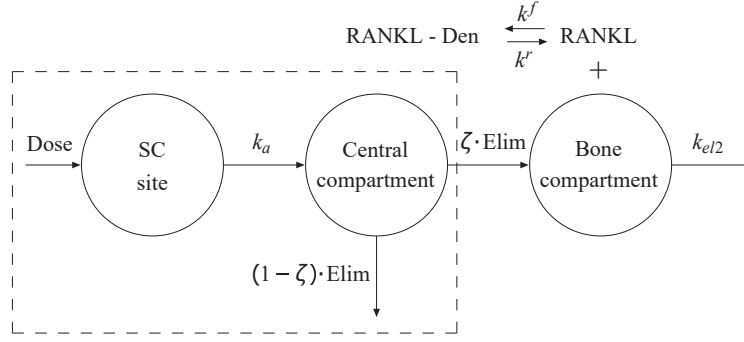


Figura 3.2 Representación esquemática del modelo PK para Dmab. El recuadro enmarcado representa el modelo PK de un solo compartimento propuesto por Marathe et al. Se observa el compartimento óseo añadido en [38]. El flujo de Dmab desde el compartimento central hacia el compartimento óseo representa el término de producción en las reacciones de unión competitiva entre RANKL, OPG y Dmab..

la que interfirió con la vía de señalización Scl-LRP5/6-Wnt. Se ha agregado el compartimento óseo porque la unión competitiva Scl-LRP5/6-Wnt tiene lugar en la superficie de los precursores de osteoblastos y solo el Rmab que llega al compartimento óseo puede interferir con esa vía de señalización.

Se ha utilizado la cinética de Michaelis-Menten para simular la respuesta de la PK mediante el ajuste de las constantes para que coincidan con los resultados encontrados en la literatura [69]. Un proceso de primer orden (k_a) gobierna la absorción del fármaco (*Dosis*) desde el sitio de inyección subcutánea (SC) hasta el compartimento central ($[SclAb]_{CC}$), donde V_c/F es el volumen del compartimento central ajustado por biodisponibilidad. La eliminación del fármaco desde el compartimento central se describe mediante una combinación de un proceso lineal de primer orden (k_{el}) y un proceso de saturación no lineal (V_{max} , K_M):

$$\begin{aligned} \frac{d[Rmab]_{CC}}{dt} &= \frac{Dose}{V_c/F} k_a e^{-k_a t} \\ &- \left[k_{el} [SclAb]_{CC} + \frac{V_{max}}{V_c/F} \frac{[SclAb]_{CC}}{K_M + [SclAb]_{CC}} \right] \end{aligned} \quad (3.152)$$

En el modelo actual, el término de eliminación que simula la excreción urinaria es solo una fracción $(1 - \zeta)$ de la cantidad eliminada del compartimento central. El resto, ζ , se asume que ingresa al compartimento óseo y se identifica con el término de producción de anticuerpos Scl durante la unión competitiva entre Scl y SclAb, $P_{SclAb_{BC}}$.

$$P_{SclAb_{BC}} = \zeta \left[k_{el} [SclAb]_{CC} + \frac{V_{max}}{V_c/F} \frac{[SclAb]_{CC}}{K_M + [SclAb]_{CC}} \right] \quad (3.153)$$

$$[SclAb]_{BC} = \frac{P_{SclAb_{BC}}}{\tilde{D}_{SclAb_{BC}} + \frac{\tilde{D}_{SclAb-Scl}}{K_{SclAb-Scl}} [Scl]} \quad (3.154)$$

Donde $\tilde{D}_{SclAb_{BC}}$ es la tasa de degradación de Rmab en el compartimento óseo.

Las constantes k_a , K_M , V_{max} y $\tilde{D}_{SclAb-Scl}$ se ajustaron para que el modelo PK coincidiera con la literatura [69]. Para ajustar el modelo PD a los resultados experimentales [70], se ajustaron ζ , $\tilde{D}_{SclAb_{BC}}$ y $K_{SclAb-Scl}$.

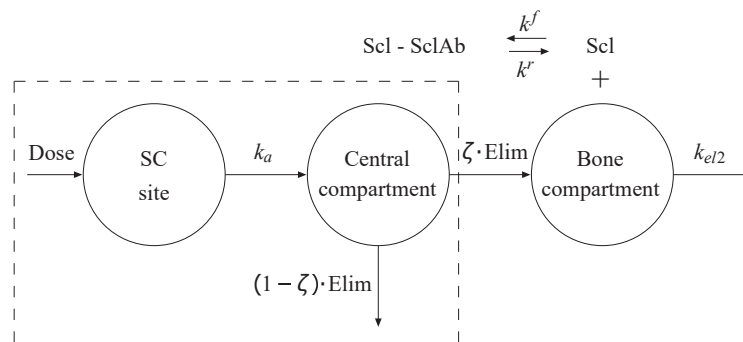


Figura 3.3 Representación esquemática del modelo PK propuesto. El recuadro enmarcado representa el modelo PK de un solo compartimento propuesto por Martin et al. [41]. Se puede ver el compartimento óseo añadido. El flujo de Rmab desde el compartimento central al compartimento óseo representa el término de producción en la reacción de unión Scl-SclAb..

4 APP DESIGNER: Interfaz gráfica

App Designer es una herramienta de desarrollo de aplicaciones que forma parte del entorno de desarrollo integrado (IDE) de Matlab. Es una interfaz gráfica de usuario (GUI) interactiva que permite a los desarrolladores de Matlab diseñar, prototipar y crear aplicaciones visualmente atractivas y funcionales sin necesidad de escribir código. En este capítulo, se brindará una introducción al entorno de App Designer y se explicarán sus principales elementos.

4.1 Introducción a App Designer

App Designer es un entorno gráfico de desarrollo que permite la creación de interfaces gráficas mediante dos componentes estrechamente relacionadas entre sí: la vista del diseño de la aplicación y la vista del código que esta detrás de la aplicación. Los usuarios pueden arrastrar y soltar los componentes visuales que prefieran como botones, cuadros de texto, gráficos, etc. Con ellos se puede crear la interfaz que dará lugar a la aplicación. A estos mismos componentes visuales, se les puede añadir un código para definir su comportamiento en la vista del código. El editor de código controla la lógica de la aplicación y este se puede relacionar con las bibliotecas y capacidades de Matlab. Los cambios realizados en ambas partes se ven reflejados en la otra, existiendo así dos maneras de modificar la aplicación.

4.1.1 Acceso

Para acceder a App Designer se debe tener Matlab instalado. Una vez abierto Matlab se puede acceder de dos maneras:

- En la ventana de inicio de Matlab, en la pestaña de 'APPS' del menú superior seleccionando "Design App".
- Escribiendo "appdesigner" en la ventana de comandos de Matlab.

4.2 Entorno de trabajo

El entorno de trabajo de App Designer se compone de diferentes elementos. Las dos vistas comentadas se ven en las figuras 4.1 y 4.2: Design View y Code View.

En este punto, se exploran las principales herramientas disponibles en el entorno de App Designer. En primer lugar, se brinda una comprensión detallada de los elementos de las dos vistas del entorno: diseño y código. Se verá en mayor detalle en qué consisten cada una de ellas. Además, se ofrece una visión general de los componentes clave de la biblioteca de componentes, que son fundamentales para la construcción de la interfaz gráfica de la aplicación. También se presenta una breve descripción de la zona del código, destacando sus partes y funciones. Por último, se definen los conceptos de función, propiedad y callbacks, imprescindibles para la comprensión de App Designer.

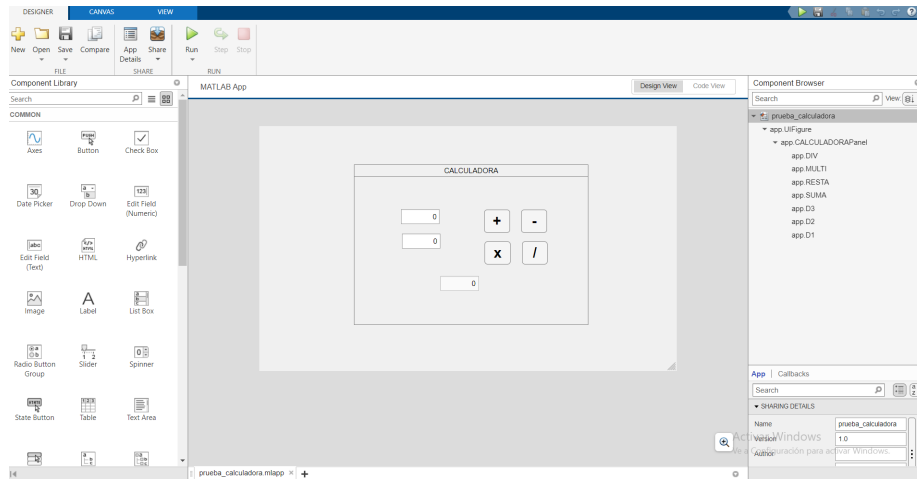


Figura 4.1 Entorno de trabajo: Design view.

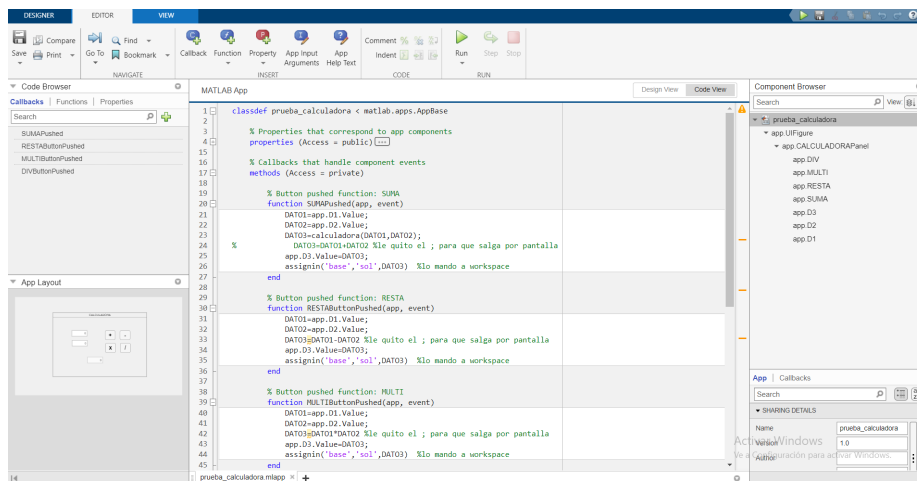


Figura 4.2 Entorno de trabajo: Code view.

4.2.1 Elementos de la vista de diseño

A continuación se describen los principales elementos de la vista de diseño o Design View:

- **Editor de diseño (Design Editor):** Es el área principal donde se pueden arrastrar y soltar los diferentes elementos visuales, ver figura 4.3 y que simulan la representación que verá el usuario.
- **Librería de componentes (Component Library):** Contiene los elementos visuales predefinidos: botones, cuadros de texto, tablas, etc (ver figura 4.4). Estos componentes son los que se pueden seleccionar y arrastrar hasta el editor de diseño y así componer la interfaz.
- **Ventana de propiedades del elemento (Button Properties):** Al seleccionar un componente en el editor de diseño se despliega en la esquina inferior derecha una ventana de propiedades de dicho elemento. En ella se puede personalizar la apariencia, el comportamiento, el valor, el nombre y otras propiedades de dicho componente.
- **Cuadrícula (Component browser):** Es una guía visual, ver figura 4.5, que muestra, ordenados por niveles, los elementos empleados en el editor de diseño con su nombre correspondiente. Esto ayuda a alinear y posicionar los elementos de manera precisa.

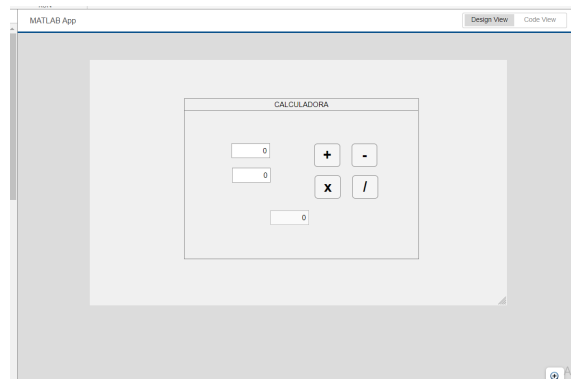


Figura 4.3 Elemento editor de diseño.

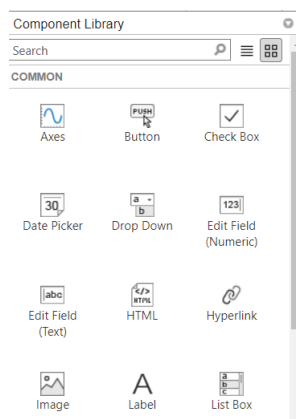


Figura 4.4 Elemento librería de componentes.

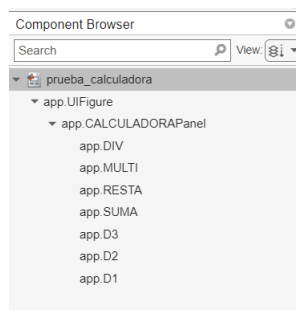


Figura 4.5 Elemento cuadrícula.

- **Barra de herramientas (Toolbar):** Esta barra (ver figura 4.6) permite el acceso rápido las herramientas más comunes dentro de la aplicación.

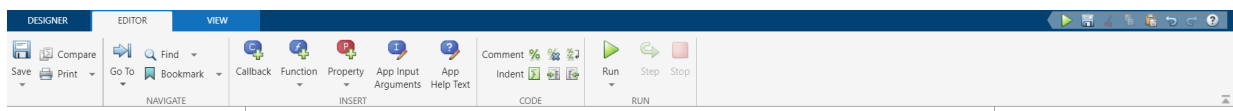
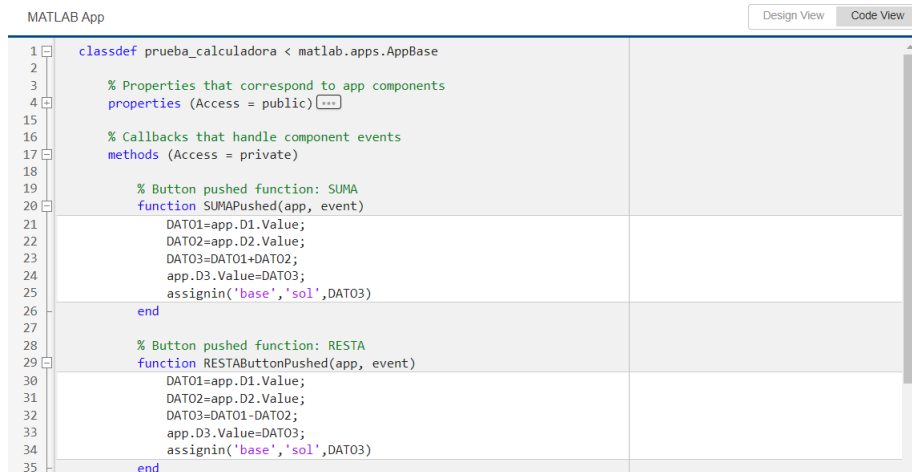


Figura 4.6 Barra de tareas.

4.2.2 Elementos de la vista de código

A continuación, se describen los principales elementos de la vista de código o Code View:

- **Editor de código (Code Editor):** Es el área principal donde se escribe y edita el código de Matlab relacionado con la interfaz (ver figura 4.7). En este código existen dos partes: la parte del código generada automáticamente por App Designer al colocar los elementos en el editor de diseño y la parte del código creada por el programador. El code editor ejerce la función de elemento principal en esta vista sobre el que se construye el código completo y las funciones de la aplicación.



```

MATLAB App
Design View Code View
1 classdef prueba_calculadora < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public) []
15
16     % Callbacks that handle component events
17     methods (Access = private)
18
19         % Button pushed function: SUMA
20         function SUMAPushed(app, event)
21             DATO1=app.D1.Value;
22             DATO2=app.D2.Value;
23             DATO3=DATO1+DATO2;
24             app.D3.Value=DATO3;
25             assignin('base','sol',DATO3)
26         end
27
28         % Button pushed function: RESTA
29         function RESTAButtonPushed(app, event)
30             DATO1=app.D1.Value;
31             DATO2=app.D2.Value;
32             DATO3=DATO1-DATO2;
33             app.D3.Value=DATO3;
34             assignin('base','sol',DATO3)
35         end

```

Figura 4.7 Elemento editor de código.

- **Code Browser:** Muestra una vista estructurada de los diferentes archivos (callbacks, funciones de utilidad, etc.) y componentes de la aplicación (ver figura 4.8) Esta ventana se encuentra en la parte superior izquierda.

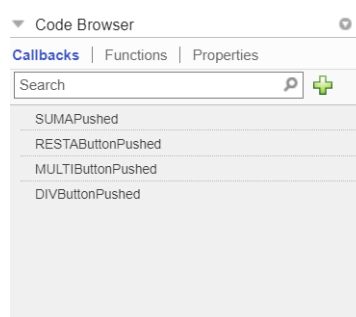


Figura 4.8 Elemento Code Browser.

- **App Layout:** Representación en miniatura del resultado de la interfaz gráfica (ver figura 4.9).
- **Component Browser y Button Properties:** Realizan la misma función que en el editor de diseño.
- **Barra de herramientas:** Al igual que en la vista de diseño, esta barra permite el acceso rápido las herramientas mas comunes dentro de la aplicación.



Figura 4.9 Elemento Layout.

4.2.3 Funcionamiento de la vista de código

El Code view es la visualización y edición del código de Matlab asociado a la aplicación. En él se puede escribir utilizando la sintaxis y las funciones propias de Matlab. Como se ha explicado anteriormente, está formado por el code editor y cuatro paneles: el Code browser, component browser, layout y button properties. El código completo que configura de aplicación está compuesto por dos partes: la parte generada automáticamente por el programa y la creada por el programador. Para distinguir ambos códigos, solo hay que ver el color de fondo del código siendo de color grisáceo para el automático y de fondo blanco el del programador [71].

Dentro del Code view existe un elemento fundamental para utilizar App Designer: los callbacks. Los callbacks son funciones esenciales que permiten controlar y responder las interacciones del usuario en la interfaz. Los callbacks sirven para programar el comportamiento de la aplicación, por ejemplo: cuando un usuario haga click en un botón de la aplicación se representen unos datos concretos, o desaparezcan otros.

El código que genera App Designer está compuesto por tres partes (ver figura 4.10):

- **Definición de las propiedades de la aplicación:** Esta parte del código es automática, y es en la que se definen las características de la aplicación.
- **Funciones:** La segunda parte del código contiene todas las funciones como callbacks y funciones de utilidad utilizadas en la aplicación. App Designer las incluye automáticamente cuando se añaden, dejando un espacio donde el programador puede incluir su parte variable del código.
- **Inicialización y creación de la aplicación y sus componentes:** Es automático y se utiliza para modificar las características de los componentes.

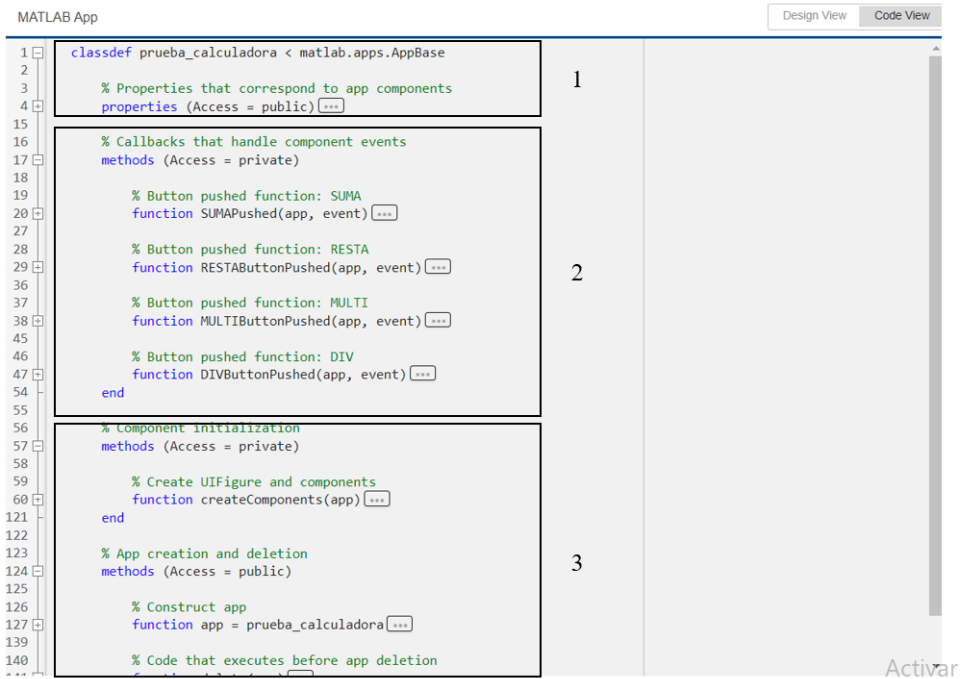


Figura 4.10 Partes del código del code editor.

4.2.4 Callbacks, propiedades y funciones

En App Designer existen tres elementos distintos pero interrelacionados utilizados para crear y controlar la interfaz.

- **Funciones:** Son bloques de código que contienen instrucciones específicas. Para poder ejecutar un código en diferentes partes de la aplicación se utilizan las funciones. En la figura 4.11 se puede ver donde se crean las funciones.

Existen dos tipos de funciones:

- Privadas, que sirven para utilizar los comandos en múltiples callbacks evitando así la repetición de código.
- Públicas, que se utilizan cuando es necesario compartir la función con el espacio de trabajo de Matlab u otras aplicaciones.

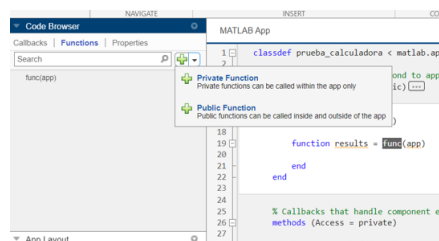


Figura 4.11 Creación de funciones privadas y públicas.

- **Callbacks:** Son funciones que se utilizan para controlar y responder a eventos que se dan en la interfaz de la aplicación. Estos eventos pueden incluir interacciones con el usuario. Algunos ejemplos de esto son: pulsar un botón, ingresar información en un cuadro de texto, marcar un check box, etc. Dichos callbacks van a permitir relacionar un evento con una función.

Cuando se produce el evento, la función correspondiente se ejecuta automáticamente. Por ejemplo, cuando se pulse un botón aparece en un campo de texto un valor concreto. Estos te permiten responder a las acciones que realiza el usuario y mantener el flujo e información. Hay varias formas de crear los callbacks, una de ellas es pulsando el botón derecho de un componente que esté en el Design view y seleccionando 'callbacks' (ver figura 4.12). Otra forma es desde el Component Browser, haciendo clic en el botón derecho sobre el nombre del componente al que se le quiere agregar el callback y luego seleccionar 'callbacks'. Una vez creado, App Designer crea automáticamente la parte del código mencionada anteriormente donde el programador tiene su área de escritura designada.

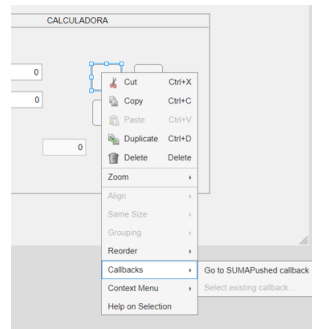


Figura 4.12 Creación de Callbacks.

- **Propiedades:** Son atributos de los componentes de la interfaz. La función de las propiedades en App Designer es almacenar datos para compartirlo entre los diferentes callbacks. Estas propiedades pueden ser privadas o públicas. Se crean desde el code browser (ver figura 4.13).

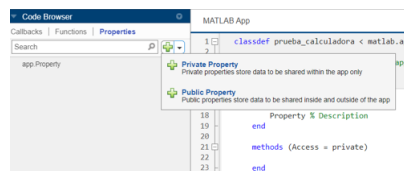


Figura 4.13 Creación de propiedad privadas y públicas.

4.2.5 Componentes de la librería de componentes

Se presentan a continuación los principales elementos de la librería de componentes (ver figura 4.14) de App Designer para la creación de una interfaz gráfica [72]. Para cada componente se detallan sus principales funciones y propiedades. Las propiedades de los elementos se pueden modificar de dos formas, como se ha adelantado anteriormente: mediante el Code View o manualmente en la vista de diseño en la ventana de propiedades del elemento.

- **Componente Label:** Las etiquetas son componentes que contienen texto estático. En sus propiedades se pueden cambiar su apariencia y comportamiento.
- **Componente Button:** Son componentes que responden cuando el usuario los presiona. En sus propiedades se pueden cambiar su apariencia y comportamiento.
- **Componente Check Box:** La casilla de verificación permite al usuario seleccionar o deseleccionar una opción. Al marcar la casilla la opción está activada y al no marcarla desactivada.
- **Componente Drop Down:** Son listas desplegables que permiten seleccionar una opción al usuario. En sus propiedades se pueden cambiar su apariencia y comportamiento.

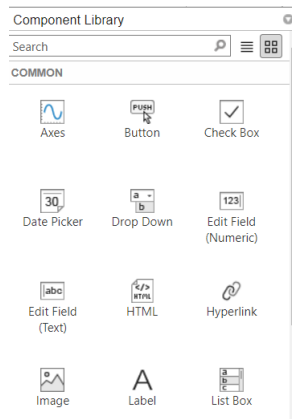


Figura 4.14 Distintos componentes de la librería..

- **Componente Edit Field (text):** Los campos de edición permiten ingresar texto al usuario, que posteriormente será procesado y utilizado en la aplicación. En sus propiedades se pueden cambiar su apariencia y comportamiento.
- **Componente Edit Field (numeric):** Los campos de edición numéricos permiten ingresar al usuario valores numérico que posteriormente serán procesados y utilizados en la aplicación. En sus propiedades se pueden cambiar su apariencia y comportamiento.
- **Componente Panel:** Los paneles son componentes que permiten agrupar otros componentes de la interfaz. En sus propiedades se pueden cambiar su apariencia y comportamiento.

4.3 Crear y ejecutar una App sencilla en App Designer

En esta guía se proporcionarán los pasos necesarios para crear una aplicación sencilla en el entorno de App Designer. La aplicación será una simulación de una calculadora y servirá como ejemplo para comprender cómo definir y relacionar los callbacks.

Para comenzar, se explicará cómo diseñar la interfaz para el usuario de la calculadora utilizando componentes como botones, cuadros de texto y etiquetas. Tras esto, se procederá a asignar los callbacks a los diferentes botones para capturar las interacciones del usuario. Seguidamente se detallarán los pasos para definir los callbacks correspondientes a las operaciones aritméticas, como suma, resta, multiplicación y división. Cada callback se encargará de realizar la operación respectiva y mostrar el resultado en el cuadro de texto.

- **Paso 1:** Crear los componentes de la interfaz gráfica: En el Design View se crean los componentes de la interfaz, pudiendo modificar su apariencia de forma interactiva. La biblioteca Component Library contiene todos los componentes y herramientas que se pueden agregar a la app. Se añaden los componentes arrastrándolos desde Component Library hasta el área de trabajo de la app (ver figura 4.15). Después, se pueden modificar las apariencias de los componente configurando propiedades en el explorador Component Browser. En el caso de la calculadora los componentes necesarios serán:
 - Edit Field (numeric): Dos campos numéricos para ingresar los dos valores con los que se van a realizar las operaciones matemáticas. Además, hay un campo más para mostrar el resultado de la operación.
 - Button: Cuatro botones para elegir las operaciones matemáticas.
 - Componente Panel: Un panel para agrupar todos los componentes.

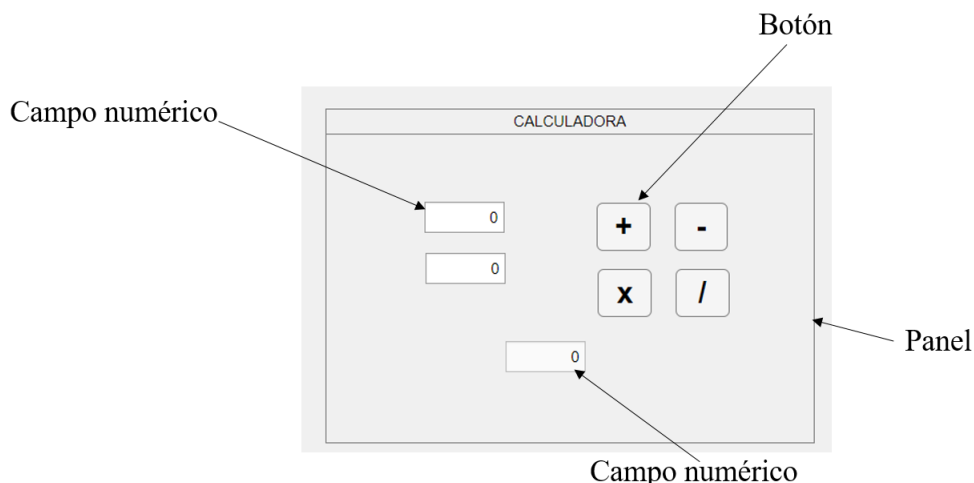


Figura 4.15 Componentes necesarios para crear la interfaz de ejemplo que simula calculadora..

- **Paso 2:** Una vez se ha diseñado la interfaz visible para el usuario, el siguiente paso es modificar el código para que la aplicación realice lo deseado. Al añadirse los componentes al Design View, se crearán también su correspondiente parte de código automático. Este código configura la apariencia de la app para que coincida con lo que se ve en el área de trabajo. Cada vez que se agrega un componente en Design View, App Designer almacena el componente como un objeto cuyo nombre se establece utilizando el formato `app.ComponentName`. Se pueden mostrar y modificar los nombres de los componentes en la app utilizando el Component Browser (ver figura 4.16). Para acceder a las propiedades de los componentes y actualizarlas desde el código de la app, se utiliza el patrón `app.ComponentName.Property`.

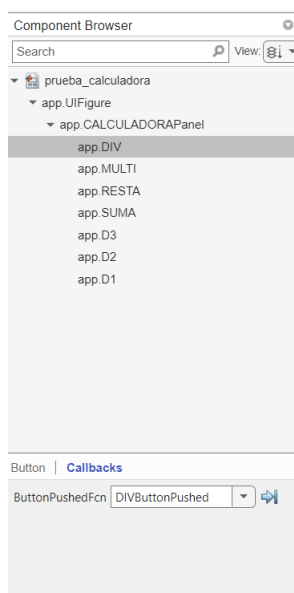


Figura 4.16 Vista del Component Browser con todos los elementos de la interfaz de la calculadora..

- **Paso 3:** Funciones callbacks: El comportamiento de la calculadora se programa utilizando los callbacks, que se ejecutan cuando el usuario lleve a cabo una interacción con a interfaz. Por ejemplo, cuando el usuario ingrese los valores de los números con lo que se quieren realizar

las operaciones, o cuando decida qué operación se quiere realizar con ellos entre suma, resta, multiplicación o división.

En el caso de la calculadora se necesita un tipo de callback concreto, el de botón presionado. Este callback, como su propio nombre indica, tiene la función de realizar las instrucciones que contenga cuando se pulse el botón específico en la interfaz. En este ejemplo existen cuatro botones, siendo las funciones de ellos las siguientes:

- **Botón suma:** Sumar los dos números ingresados por el usuario en los campos numéricos de la interfaz y mostrar el resultado de la operación en el campo numérico que se encuentra más abajo.
- **Botón resta:** Restar los dos números ingresados por el usuario en los campos numéricos de la interfaz y mostrar el resultado de la operación en el campo numérico que se encuentra más abajo.
- **Botón multiplicación:** Multiplicar los dos números ingresados por el usuario en los campos numéricos de la interfaz y mostrar el resultado de la operación en el campo numérico que se encuentra más abajo.
- **Botón división:** Dividir los dos números ingresados por el usuario en los campos numéricos de la interfaz y mostrar el resultado de la operación en el campo numérico que se encuentra más abajo.

Una vez conocidos y definidos los callbacks necesarios, se crean como se ha explicado en el punto 4.2.4. Tras esto aparecerá en el Code View su correspondiente espacio para la programación editable (ver figura 4.17). En los cuatro callbacks el procedimiento a seguir será el siguiente:

1. Recoger los valores ingresados por el usuario en los dos campos numéricos y asignarle un nombre.
2. Realizar la operación correspondiente y asignarle un nombre.
3. Devolver por el campo numérico la solución a la operación igualando el valor de este al de la variable definida en el paso anterior.

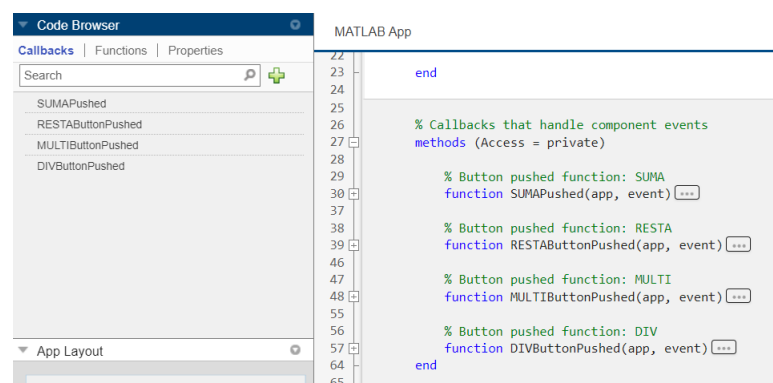


Figura 4.17 Visualización de los cuatro callbacks en el Component Browser y sus partes de códigos correspondientes en el Code View..

A continuación se muestra el código del callbacks SUMA:

```
1 % Callbacks that handle component events
2 methods (Access = private)
```

```
3
4     % Button pushed function: SUMA
5     function SUMAPushed(app, event)
6         DAT01=app.D1.Value; % Recogida del primer número
7         DAT02=app.D2.Value; % Recogida del segundo número
8         DAT03=DAT01+DAT02; % Se suman ambos y se le da el nombre de
9             DAT03
10        app.D3.Value=DAT03; % Se iguala el valor del campo numérico D3
            a la variable DAT03
        assignin('base','sol',DAT03)
```

Los otros tres códigos siguen la misma dinámica. Una vez completados estos tres pasos se consigue una interfaz lista para la interacción con el usuario sin requerir programación por parte de este. El ejemplo de la calculadora es simple e ilustrativo para conocer los elementos principales de App Designer, pudiendo comprender fácilmente esta herramienta que proporciona Matlab.

5 Calculadora del riesgo de fractura

En este capítulo se desarrolla el diseño de una interfaz gráfica mediante App Designer. Esta interfaz funciona como una calculadora del riesgo de fractura. Su objetivo es conocer la evolución de la densidad ósea, la fracción volumétrica de hueso, el daño y la probabilidad de fractura en función de los datos de entrada de la paciente y los datos del tratamiento, utilizando el modelo matemático desarrollado en el capítulo 3.

5.1 Justificación y objetivo

Gracias al modelo matemático expuesto en el capítulo 3 se puede simular la remodelación ósea en pacientes con osteoporosis postmenopáusica (PMO).

Sin embargo, el manejo del modelo requiere un conocimiento de programación, los cuales dificultan su utilización por parte de profesionales de la salud u otras personas que no estén familiarizadas con su funcionamiento interno.

Como consecuencia, se propone en este capítulo diseñar una interfaz sencilla que actúe como puente entre el modelo matemático y los usuarios finales. Dicha interfaz permite al usuario ingresar los datos de entrada requeridos para el modelo, como la edad de la menopausia, la edad actual, el peso y otros parámetros relevantes de su historial clínico. Una vez que los datos de entrada sean introducidos por el usuario en la interfaz, estos son automáticamente enviados y utilizados por el modelo matemático para realizar los cálculos necesarios y generar soluciones.

Estas soluciones podrían incluir estimaciones de la densidad ósea futura, el riesgo de fractura, la eficacia de ciertos tratamientos y otros resultados relevantes para el diagnóstico de los pacientes. En este caso concreto las soluciones obtenidas por medio de la interfaz son la evolución de la densidad ósea, fracción volumétrica de hueso y daño desde los 20 años de edad hasta 10 años después del tratamiento.

Este enfoque tiene como objetivo agilizar y simplificar el proceso de obtención de resultados del modelo, brindando a las personas que no tienen manejo con él, una herramienta intuitiva y fácil para obtener información futura sobre los parámetros de remodelación y tratamiento.

5.2 Introducción

Se lleva cabo el diseño de una interfaz compuesta por dos secciones:

- **Primera sección, 'Formulario':** permite la introducción de información fundamental acerca de la paciente y los detalles del tratamiento futuro que se pretende analizar mediante la implementación del modelo matemático del capítulo 3. El propósito principal es evaluar el impacto de dicho tratamiento en una persona específica. Una vez que se ha ingresado la

información requerida se pulsa el botón que activa el proceso de generación de resultados. Las salidas de esta primera parte de la interfaz son las evoluciones a lo largo del tiempo de la densidad ósea, fracción volumétrica de hueso y daño.

- **Segunda sección, 'Pronóstico de Probabilidad de Fractura'**: permite, una vez completada la primera parte del formulario, calcular y mostrar la probabilidad de fractura. Para realizar este cálculo, es necesario proporcionar los datos correspondientes y, nuevamente, se utiliza un botón para generar la solución deseada. La salida de esta segunda parte es la probabilidad de fractura a lo largo del tiempo.

En este capítulo, se desarrolla la creación de la interfaz de la siguiente manera. En primer lugar, en el punto 5.3, se presentan los datos de entrada y salida de la interfaz. A continuación, en el punto 5.4, se describe el proceso de diseño de la interfaz en el entorno de App Designer, incluyendo la conexión de dicha interfaz con el modelo matemático ya existente. Posteriormente, en el punto 5.5, se aborda el cálculo del parámetro k_5 , el cual fue mencionado previamente en el punto 3.8.3 y es necesario para adaptar el modelo a cada caso concreto de paciente en función del T-score. Finalmente en el punto 5.6 se muestra un ejemplo del funcionamiento de la misma.

5.3 Datos de entrada y salida

Se presentan a continuación los datos de entrada y salida de la interfaz. Los datos de entrada se van a dividir en tres grupos: datos de perfil de la paciente, datos del tratamiento y datos de probabilidad de fractura.

- Datos de perfil de la paciente:
 - Nombre completo
 - Edad actual
 - Edad de la menopausia
 - Peso
 - Sexo
 - Fecha de nacimiento
- Datos del tratamiento:
 - Fármaco: Se evalúan dos opciones de fármaco, el denosumab y el romosozumab.
 - Zona: Se evalúan dos tipos de hueso: fémur y vertebra.
 - Dosis: Miligramos de fármaco inyectados en cada inyección.
 - Lapso entre inyecciones/Frecuencia: Número de días entre dos inyecciones de fármaco.
 - Número de Inyecciones: Será fijo con un número de diez inyecciones en todo el tratamiento.
 - T-Score: Es una medida que se utiliza para evaluar la densidad ósea. Representa la desviación estándar de la densidad ósea de un individuo en comparación con la densidad ósea promedio de una población del mismo sexo y edades comprendidas entre 20-39 años. Se explica más en detalle en el apartado 5.5.1
- Datos de probabilidad de fractura:
 - Nombre

- Carga aplicada
- Zona: Se evalúan dos tipos de hueso: fémur y vertebra.
- Tensión: si el hueso se evalúa a tracción o a compresión.

Es importante resaltar que en la interfaz, que se puede ver más adelante en el punto 5.4, también se incluye una sección dedicada a los parámetros considerados por la herramienta FRAX, presentada en el punto 1.1.2. Aunque estos datos no se utilizan actualmente, se han incorporado en la interfaz con la finalidad de facilitar posibles mejoras y futuros trabajos.

Los datos de salida de la interfaz son las evoluciones de los siguientes parámetros a lo largo de tiempo desde la homeostasis, pasando por el tratamiento y 10 años después de este:

- **Densidad:** Evolución de la densidad de la paciente a lo largo del tiempo.
- **Fracción volumétrica de hueso:** Evolución de la fracción volumétrica de hueso de la paciente a lo largo del tiempo.
- **Daño:** Evolución del daño de la paciente a lo largo del tiempo.
- **PF:** Probabilidad de fractura a lo largo del tiempo.

5.4 Diseño y desarrollo de la interfaz

5.4.1 Diseño en App Designer

A continuación, se presenta en detalle el desarrollo de la interfaz, utilizando los principios y conocimientos expuestos en el capítulo 4 como base fundamental.

El proceso de diseño de la interfaz en App Designer comienza en la vista Design View. En esta vista, se seleccionan y colocan los componentes adecuados en la interfaz, teniendo en cuenta el tipo de datos que se requieren para capturar la información personal de la paciente.

Para recopilar los datos personales de la paciente, se pueden utilizar diferentes tipos de componentes (explicados en el punto 4.2.5) en la interfaz:

- Campos de texto: Se utilizan para ingresar información como nombre, apellido, etc. Los campos de texto permiten al usuario escribir texto.
- Campos numéricos: Se utilizan para ingresar información como la edad actual, la edad de la menopausia, el peso, etc. Los campos numéricos permiten al usuario escribir solo números.
- Botón: En este caso, se agrega un botón de "Finalizar" en la primera sección de la interfaz para indicar la finalización del proceso de ingreso de datos y activar el modelo matemático. Del mismo modo, se añade un segundo botón para calcular la probabilidad de fractura (PF) en la segunda sección.
- Casillas de verificación: Se utilizan para recopilar información en forma de opciones binarias, como sí/no o verdadero/falso. Se utilizan para seleccionar el tipo de fármaco, zona de estudio y tensión.
- Paneles: Se utilizan para organizar y estructurar los datos agrupando componentes relacionados y facilitando así la presentación de la interfaz. Se utilizarán los siguientes paneles:
 - Panel de datos personales
 - Panel de datos de tratamiento
 - Panel de datos del FRAX (ya se ha comentado que en este trabajo no se utiliza)

Figura 5.1 Interfaz Calculadora: Elementos de la vista de diseño de la interfaz.

- Panel de datos de pronóstico de PF.

Una vez que se han seleccionado y colocado los componentes en la vista de diseño de App Designer (ver figura 5.1), el siguiente paso es asignar nombres descriptivos a cada uno de los elementos en el navegador de componentes. El navegador de componentes muestra una lista organizada de todos los elementos presentes en la interfaz, según los paneles previamente definidos, ver figura 5.2.

Para facilitar el desarrollo posterior del código de la aplicación, los elementos deben tener nombres significativos. Estos nombres sirven como identificadores distintivos que permiten al programador identificar y acceder a las propiedades y eventos de cada componente en el código.

A continuación, tras configurar los nombres y las propiedades de cada componente en la interfaz de App Designer, el siguiente paso es determinar los callbacks correspondientes a cada componente. Los callbacks son funciones o fragmentos de código que se ejecutan en respuesta a eventos particulares, como cuando un componente cambia de valor o un botón se pulsa.

En el caso de este código, se utiliza un callback esencial, el callback de botón presionado. Este callback se activa cuando se presiona un botón en la interfaz. En el caso específico de la interfaz de estudio, en la primera sección de la interfaz, cuando el usuario presiona el botón de "Finalizar", se considera que se ha completado el proceso de relleno de datos del paciente y se inicia la ejecución del modelo matemático. Del mismo modo, en la segunda sección, una vez se pulse el botón "PF" se muestran las soluciones.

Las acciones que se llevan a cabo tras pulsar el botón de la primera sección:

- **Fin del relleno de datos:** Al pulsar el botón los datos introducidos serán enviados al modelo para dar las salidas necesarias.

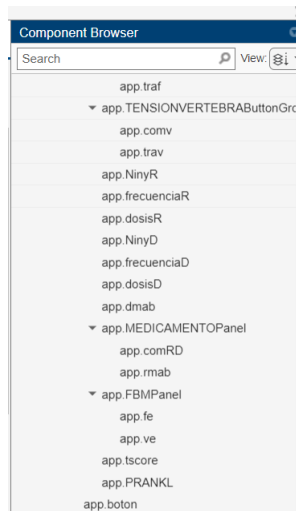


Figura 5.2 Vista del Component Browser con los nombres de los componentes de la interfaz.

- **Inicio del modelo matemático:** Una vez que los datos se consideren correctos, se inicia el proceso del modelo matemático, utilizando los valores ingresados como entradas.
- **Vaciar los campos de entrada:** Después de que el modelo se haya iniciado, se borran los campos de ingreso de texto y números, de manera que estén listos para ser rellenados nuevamente con otros datos en futuras interacciones. Esto brinda una experiencia más intuitiva al usuario y evita confusiones o errores al ingresar nuevos datos.

Una vez se ha definido que el callback 'botonButtonPushed' se procede a desarrollarlo:

1. Se abre la ventana de código en App Designer haciendo clic en el botón "Code View" en la pestaña "Design", ver figura 5.3.

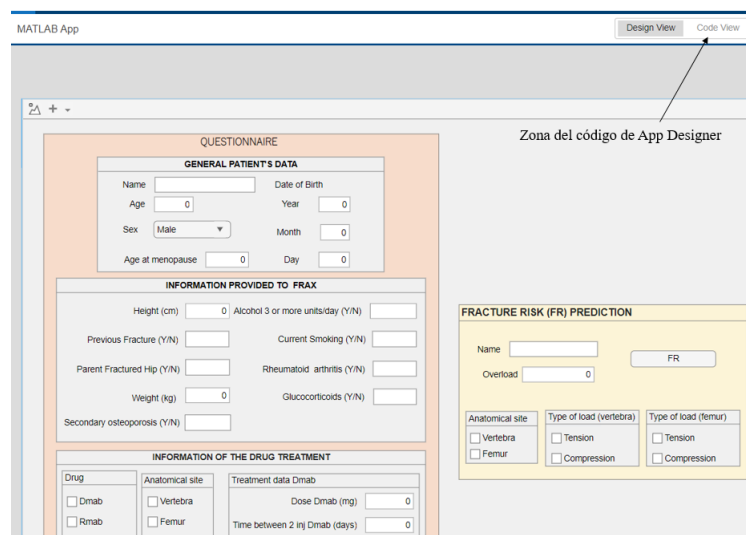
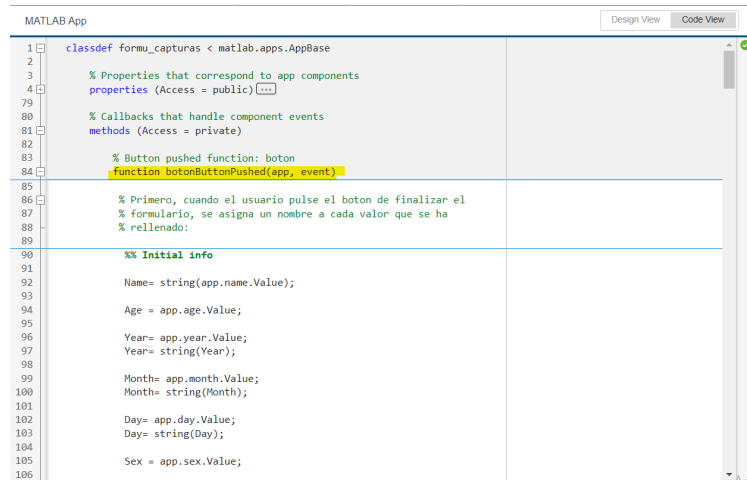


Figura 5.3 Vista de diseño.

2. Dentro de la función de callbacks del botón finalizar (ver figura 5.4), se programa el código que se ejecuta cuando se pulse el botón:



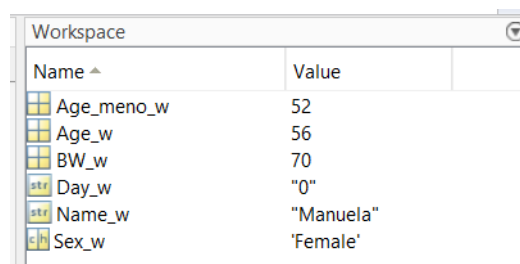
```

1 classdef formu_capturas < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public) (...)
79
80     % Callbacks that handle component events
81     methods (Access = private)
82
83         % Button pushed function: boton
84         function botonButtonPushed(app, event)
85
86             % Primero, cuando el usuario pulse el boton de finalizar el
87             % formulario, se asigna un nombre a cada valor que se ha
88             % rellenado:
89
90             %% Initial info
91             Name= string(app.name.Value);
92
93             Age = app.age.Value;
94
95             Year= app.year.Value;
96             Year= string(Year);
97
98             Month= app.month.Value;
99             Month= string(Month);
100
101             Day= app.day.Value;
102             Day= string(Day);
103
104             Sex = app.sex.Value;
105
106

```

Figura 5.4 Vista de código.

- Asignar valores a las variables de la interfaz: El primer paso del código consiste en obtener los valores de los elementos de la interfaz que necesitan. Por ejemplo, en el caso del campo llamado "name" que recoge el nombre ingresado por el usuario, se puede obtener su valor utilizando `app.name.Value`. Luego, se puede asignar este valor a una variable llamada "Name" utilizando `Name = string(app.name.Value)`. Este proceso se hace con todas las variables que recoge el formulario, para almacenar toda la información.
- Enviar las variables al workspace de Matlab: Después de haber obtenido los valores de las variables de la interfaz, se envían al workspace de Matlab (ver figura 5.5), para poder trabajar con ellas en el modelo matemático. Esto se puede lograr utilizando la función `assignin('base', 'Name_w', Name)`, donde "Name_w" es el nombre de la variable en el workspace y "Name" es el valor que se le está asignando. Este proceso se vuelve a realizar con todas las variables necesarias para realizar el modelo. Algunas variables informativas no necesitarán ser enviadas al workspace.



Name	Value
Age_meno_w	52
Age_w	56
BW_w	70
Day_w	"0"
Name_w	"Manuela"
Sex_w	'Female'

Figura 5.5 Elementos de la vista de código.

- Definir variables condicionales: Se definen variables cuyo valor dependerá de las opciones seleccionadas en los checkboxes. Por ejemplo, en el caso del checkbox que marca el tipo de tratamiento (denosumab o romosozumab), se ha definido una variable interna dentro del modelo matemático que en función del valor que ingrese el usuario, se selecciona el tratamiento automáticamente sin tener que hacerlo manualmente dentro del modelo. Para esto se han usado condicionales.
- Una vez se han recopilado todos los valores de las variables y se tienen en un sitio donde poder manejarlas, el paso final es simular el modelo matemático para conseguir los resultados. Esto se ha realizado modificando el modelo inicial para hacerlo compatible con el

formulario. Con ello, solo llamando a tres funciones en el código se desarrolla todo el modelo. Esas tres funciones son:

- *valork5*: calcula el parámetro k_5 definido en el punto 3.8.3. Ese parámetro modela el incremento de RANKL, para poder así tener en cuenta la PMO en el modelo matemático. Se detalla la obtención de k_5 en el punto 5.5
- *fu_modelparameters*: define los parámetros del modelo.
- *Main_bone_model_SclAb*: aquí está desarrollado el modelo matemático.

Estas funciones, que se explican más adelante en el punto 5.4.2, son el nexo entre el modelo y la interfaz. En este punto no es necesario saber cual es su contenido, solo que son necesarias para el funcionamiento de la interfaz.

De manera análoga, una vez definido el callback 'boton2ButtonPushed' para la parte 2 de la interfaz se procede a desarrollarlo:

- Ejecución de los datos de la Parte 1 de la interfaz: Una vez introducido el nombre de la paciente en la sección Pronosticador de Probabilidad de Fractura, los datos correspondientes a la Sección 1 (incluyendo densidad ósea, fracción volumétrica de hueso, etc.) se activan automáticamente para su disponibilidad inmediata y posterior utilización.
- La información relativa a la carga aplicada (overload en la figura 5.1) para su posterior uso y se almacena en el workspace.
- Definir variables condicionales: gracias a ellas se conoce el tipo de hueso y si el escenario es a tracción u a compresión, en función de la selección del usuario.
- Una vez se ha recopilado toda la información necesaria para el pronóstico de PF, el paso final es ejecutar la función de Matlab *remainig_life_PYR* que realiza el estudio de la probabilidad de fractura. Más adelante en el punto 5.4.2 se ven las entradas a esta función.

Después de completar el código en App Designer en base a los dos callback principales ("*botonButtonPushed*" y "*boton2ButtonPushed*") el formulario está listo para ser utilizado. No obstante, para lograr una conexión eficiente entre el modelo y la interfaz, se requieren algunos ajustes adicionales fuera del entorno de programación de App Designer. Estos ajustes permiten una comunicación entre el modelo y la interfaz que se explica a continuación.

5.4.2 Puente entre el modelo matemático y la interfaz gráfica

Como se ha mencionado previamente, la interfaz proporciona una forma sencilla de ingresar los datos necesarios para el modelo y obtener la información deseada por parte del usuario, sin requerir conocimiento sobre los detalles internos del funcionamiento del modelo. Se puede considerar la interfaz como una "caja negra" en la que solo es necesario proporcionar las entradas, presionar un botón y observar las soluciones generadas por el modelo en la pantalla.

No obstante, para lograr este nivel de integración entre la interfaz y el modelo matemático, se requiere realizar ciertas modificaciones en el modelo original. Estas modificaciones se llevan a cabo con el propósito de establecer la conexión entre los datos de entrada provenientes de la interfaz y los datos que el modelo matemático utiliza internamente. Las modificaciones son de dos tipos: conversión del script de Matlab del modelo en funciones con entradas y salidas específicas, y asignación de variables ingresadas por el usuario a variables definidas en el modelo.

Dentro del modelo matemático, se encuentran numerosas funciones y scripts de Matlab que contienen una gran cantidad de información relevante. Sin embargo, existen algunos archivos específicos que son especialmente importantes cuando se trata de establecer la conexión entre la interfaz y el modelo.

Destacan los tres siguientes:

- **Command_window**

Este archivo permite un acceso rápido a las variables del modelo que más varían para los diferentes casos de estudio. Estas variables incluyen la edad de la paciente, la edad de la menopausia, el tiempo de estudio futuro y la fracción volumétrica de hueso. Al utilizar el *Comand_window*, es posible modificar estos parámetros de manera conveniente, evitando tener que navegar a través del código del modelo y buscar los lugares específicos donde se deben realizar los cambios. En lugar de eso, se proporciona un lugar de ingreso rápido y accesible.

- **Model_Parameters**

El archivo *fu_modelparameters* es un componente fundamental de la rutina, ya que es donde se definen de manera paramétrica, utilizando la sintaxis de Matlab, todas las variables necesarias del modelo. En este archivo, se establecen y asignan los valores iniciales de las variables relevantes. Aquí se deben asignar los nuevos valores ingresados en el formulario.

- **Main_bone_model_SclAb**

Es el archivo principal y sirve como punto central de control y coordinación, donde se gestionan las diferentes partes del programa y se realizan las llamadas necesarias para avanzar en el desarrollo del modelo matemático.

En primer lugar, se llama al archivo *Comand_window*, el cual se creó para facilitar la interacción y la selección de opciones del programa. Este archivo permite elegir diferentes casos de carga y almacenar los valores de las variables seleccionadas utilizando los comandos de Matlab *setappdata* y *getappdata*.

A continuación, se obtienen los valores de los parámetros del modelo utilizando la función *fu_modelparameters*, que devuelve una estructura con dichos parámetros.

Se definen los intervalos de tiempo. En general, se considera un intervalo de tiempo para la homeostasis y otro para la perturbación, que puede incluir factores como la edad, la enfermedad o el desuso.

Después de definir los tiempos, se llama a la función *fu_loadCase*, donde se selecciona y desarrollan los diferentes casos de carga, como por ejemplo, solo el efecto de la edad, o el efecto de la edad y la PMO, o el efecto de la edad más la PMO y el tratamiento correspondiente contra ella. A continuación, la función *steady_state_calc_withspsurf*, inicializa las poblaciones utilizando las condiciones iniciales almacenadas en los diferentes archivos que se encuentran en la rutina. Luego, se inicializa el vector *CellsVector*, que contiene además de las poblaciones celulares, los parámetros a estudiar, como el daño, la fracción de ceniza, la densidad o la fracción volumétrica. Este vector se utiliza para resolver las ecuaciones, algunas de las cuales son algebraicas y otras son ecuaciones diferenciales. Las ecuaciones diferenciales se resuelven utilizando el método de Euler, que utiliza las derivadas obtenidas de los artículos de referencia. Además, se encuentra la función *fu_micromech*, que proporciona las propiedades mecánicas como el módulo de Young, las deformaciones y algunas derivadas necesarias.

Una vez que se han conocido los archivos fundamentales del modelo matemático, se adquieren los conocimientos necesarios para comprender los pasos a seguir en la conexión. Dado que los archivos *Main_bone_model_SclAb.m* y *fu_modelparameters* son los más relevantes y contienen la información utilizada desde el formulario, se procede a convertir estos archivos en funciones, tomando como parámetros de entrada las variables del formulario. Para llevar a cabo este proceso, es necesario identificar dónde se encuentran las variables cuyo valor debe ser reemplazado, y luego igualarlas al valor proveniente del formulario. Quedando así tres funciones:

- *ModelParameter = fu_modelparameters (Fbm_t0,P_RANKL_d_PMO,BW,Dosis_Dmab,Frecuencia_Dmab,Ninyecciones_Dmab,Dosis_Rmab,Frecuencia_Rmab,Ninyecciones_Rmab,k5_m);*
 - *Fbm_t0*: Fracción volumétrica de hueso inicial.
 - *P_RANKL_d_PMO*: Producción de RANKL durante la PMO. Este parámetro es k_5 . Se añadió inicialmente para introducir un valor concreto sin necesidad de conocer el T-score. Finalmente es sustituido por T-score que se convierte en k_5 gracias a la matriz 3d que se explica en el punto 5.5.3. Se mantiene esta entrada al formulario por si de ayuda en un futuro.
 - *BW*: Peso de la paciente
 - *Dosis_Dmab, Frecuencia_Dmab y Ninyecciones_Dmab*: Dosis, lapso de tiempo entre inyecciones y número de inyecciones para el tratamiento de denosumab.
 - *Dosis_Rmab, Frecuencia_Rmab y Ninyecciones_Rmab*: Dosis, lapso de tiempo entre inyecciones y número de inyecciones para el tratamiento de romosozumab.
- *Main_bone_model_SclAb(Name,Age,Age_meno,ModelParameter,Fbm_t0,comb);*
 - *Name*: Nombre de la paciente. Necesario para poder guardar un archivo .mat con toda la información sobre los parámetros que describen su caso.
 - *Age*: Edad actual
 - *Age_meno*: Edad de la menopausia
 - *ModelParameter*: Vector con las constantes del modelo.
 - *Fbm_t0*: Fracción volumétrica de hueso inicial
 - *comb*: Variable que define el tipo de tratamiento.
- *Remainig_life_PyR(Nombre,CellsVector,TimeVector,compresion_w,traccion_w,Overload_w);*
 - *Nombre*: Nombre de la paciente.
 - *CellsVector*: Vector que contiene los parámetros de la remodelación ósea: densidad, daño, fracción volumétrica de hueso, fracción de ceniza etc.
 - *Timevector*: Vector de tiempo.
 - *Compresión*: Variable que define si es a compresión.
 - *Tracción*: Variable que define si es a tracción.
 - *Overload*: Valor de la sobrecarga aplicada.

5.5 Cálculo del parámetro k_5

5.5.1 Introducción

En el diseño y desarrollo de la interfaz (ver punto 5.4) en App Designer, se mencionó la necesidad de crear tres funciones para la implementación del modelo matemático. De ellas, dos han sido explicadas detalladamente en el punto anterior, 5.4.2, y la tercera se aborda en este apartado.

Esta última función tiene como objetivo calcular el parámetro k_5 , que representa la primera pendiente de la curva que modela el incremento de la producción de RANKL debido a la osteoporosis en función de los años transcurridos desde la menopausia (YSM). Dicho parámetro se puede relacionar con el T-score.

El T-score es un valor estadístico que se utiliza en medicina para evaluar la densidad mineral ósea de una persona al compararla con la densidad ósea de una población de referencia. Un T-score por encima o debajo de 0 indica mayor o menor densidad ósea, respectivamente. Es por ello que se utiliza para diagnosticar la osteoporosis y evaluar el riesgo de fracturas [73].

La relación matemática entre el T-score y la densidad es la siguiente:

$$\text{T-score} = \frac{\rho_l - \rho_m}{\sigma} \quad (5.1)$$

Donde ρ_l representa la densidad a evaluar, ρ_m es la densidad media de una población de mujeres de la misma raza sanas y con edades comprendidas entre 20-39 años, y σ es la desviación típica de esa población. Siendo la densidad promedio de $0.47 \frac{\text{g}}{\text{cm}^2}$ y σ de $0.04 \frac{\text{g}}{\text{cm}^2}$ [74].

La Organización Mundial de la Salud (OMS) establece los siguientes criterios para diagnosticar la osteoporosis basados en el T-score [75]:

- Normal: T-score igual o superior a -1.
- Osteopenia (baja densidad ósea): T-score entre -1 y -2.5.
- Osteoporosis: T-score igual o inferior a -2.5.

El T-score es un valor que se obtiene a partir de una densitometría ósea. Es una medida que el paciente puede conocer y proporcionar en el formulario, por ello, es una entrada requerida. Gracias a la relación matemática con la densidad, se puede relacionar el T-score con el parámetro k_5 , existiendo entonces una manera de determinar el valor de k_5 conociendo la densidad y el valor T-score de la paciente.

5.5.2 Objetivo

El objetivo es tener una base de datos que permita determinar el valor de k_5 , necesario para el modelo matemático, cuando se introduce un T-score específico en el formulario.

Para establecer esa relación entre el T-score y el k_5 , se utiliza la densidad como intermediario ya que es un valor que se puede obtener a través del modelo del capítulo 3, mientras que, no se puede calcular directamente el valor de k_5 a partir de este.

Gracias al valor de k_5 , se modela el incremento de producción de RANKL, teniendo así en cuenta en el modelo el efecto de la menopausia y, con ello, la osteoporosis, tal como se explicó en el punto 3.8.3. Para determinar este parámetro se crea una base de datos que permite establecer un valor de k_5 para cada valor de T-score dado, para una edad y años desde la menopausia (YSM) determinados, desarrollado esto en el punto 5.5.3.

Por último, en el punto 5.5.4 se lleva a cabo la integración de esta base de datos, con el propósito de ser utilizada, en la interfaz. Gracias a ella, al proporcionar un valor T-score de la paciente, en el formulario, es posible obtener el valor correspondiente de k_5 .

5.5.3 Matriz 3D

Para poder obtener todas las posibles combinaciones del valor de T-Score para distintos datos de las pacientes, se propone el siguiente enfoque: Se calcula la densidad en función de la edad de la paciente, los años transcurridos desde la menopausia y el valor de k_5 (ver ecuación 5.2). Para ello, se crea una base de datos que contiene las densidades utilizando un array de tres dimensiones, conocida como matriz 3D. La densidad y el T-score se relacionan mediante la ecuación 5.1.

$$\rho = F(\text{edad}, \text{YSM}, k_5) \quad (5.2)$$

Se define la base de datos en base a diferentes hoja de datos. La hoja de datos, ver figura 5.6, es una matriz con N filas y M columnas, que representa los valores de densidad ósea de una paciente con

edad determinada para distintos valores de YSM y k_5 .

Las filas representan los valores del parámetro k_5 , mientras que, las columnas, la diferencia entre la edad actual y la edad de la menopausia, lo que equivale al parámetro YSM. Cada celda de esta matriz está rellena con el valor de densidad correspondiente para cada combinación de estos parámetros.

El número de hojas corresponde al número de casos en función de la edad actual de la paciente. Es decir, partiendo de una edad de paciente de 50 años, las hojas de datos corresponderían a edades de las pacientes de 50 a 80 años. Siendo entonces el número de hojas igual a 31.

Para rellenar las hojas de datos de las diferentes edades de la paciente se realizan simulaciones. En dichas simulaciones se estudian las combinaciones de edad, YSM y k_5 a través del modelo matemático. La salida a las simulaciones es la densidad en cada caso.

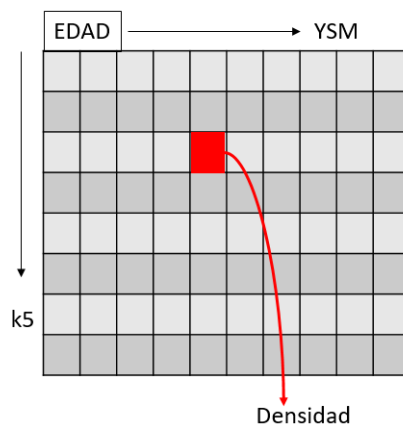


Figura 5.6 Esquema de las Hojas de datos.

A continuación, se describen los pasos generales para obtener la matriz 3D, proporcionando una visión inicial. Posteriormente, se detalla el proceso.

1. **Obtener la información necesaria:** En primer lugar, se debe buscar y recopilar la información necesaria para las simulaciones que se van a realizar. Esto implica la búsqueda de la densidad media de una población de mujeres de la misma raza, sanas y con edades comprendidas entre 20-39 años, mencionada en el punto 5.5.1. Este valor permite conocer la fracción volumétrica de hueso inicial (fbm) y la tensión de equilibrio de homeostasis correspondiente a esa densidad. Una vez conocidos ambos parámetros, se podrá calcular las condiciones iniciales para las simulaciones.
2. **Adaptar el modelo matemático:** Una vez que se tiene la información inicial, se adapta el modelo matemático del capítulo 3, para que pueda recibir y utilizar correctamente los valores de los datos que van a variar en las diferentes simulaciones. Estos valores son la edad de la paciente actualmente, YSM y k_5 .
3. **Archivo de Matlab para realizar las simulaciones:** Para llevar a cabo las simulaciones de los diferentes casos de estudio, es necesario crear un archivo en Matlab que realice las siguientes tareas: definir las dimensiones de la matriz 3D, establecer los valores correspondientes a los tres parámetros variables (edad, YSM y k_5) y llamar al modelo para su ejecución en función de los

diferentes casos de estudio. Una vez que el modelo ha simulado un caso concreto, se recopila la densidad resultante y se almacena en la hoja de datos correspondiente.

- 4. Conversión de matriz 3D de densidad a T-score:** Por último, la matriz 3D de densidad se transforma en matriz 3D de T-score utilizando la ecuación 5.1 que relaciona la densidad ósea con el T-score.

Pasos previos

PASO 1:

Para obtener la densidad promedio en mujeres con edades comprendidas entre 20 y 39 años, se llevó a cabo una búsqueda exhaustiva de información. Se consultaron fuentes relevantes como artículos científicos, y se encontró un artículo específico [74] que proporcionaba el valor de densidad necesario para calcular la fracción volumétrica de hueso y la tensión de equilibrio. La densidad promedio tiene un valor de $0.47 \frac{\text{g}}{\text{cm}^2}$.

A partir de este valor de densidad y las curvas (ver figuras 5.7 y 5.8) de fracción volumétrica y densidad de homeostasis, se obtienen los valores iniciales de fracción volumétrica de hueso y tensión de equilibrio (ver tabla 5.1):

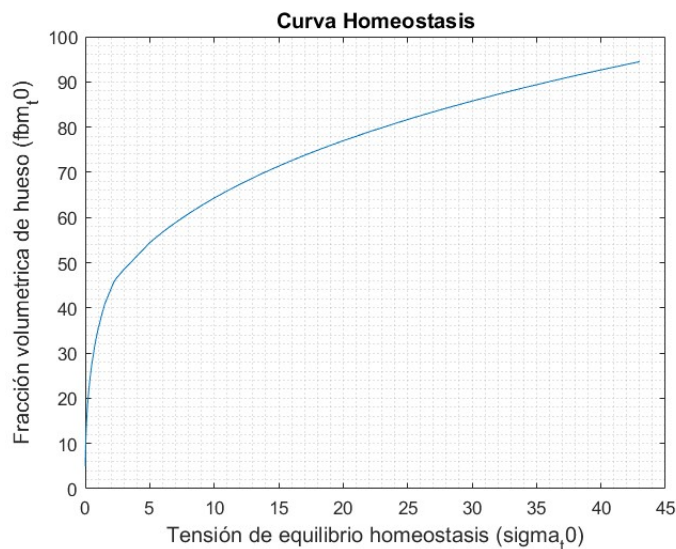


Figura 5.7 Curva de homeostasis fbm.

Tabla 5.1 Valores de los parámetros iniciales: Densidad, Tensión y fbm.

Parámetro	Valor
Densidad	0,47 g/cm ³
Fracción volumétrica de hueso	21,84 %
Tensión	-0.2834 MPa

Para acabar este punto, se deben determinar las condiciones iniciales de equilibrio de homeostasis. Para conseguirlas, se ejecuta una simulación de homeostasis, es decir, en equilibrio, con los valores de fbm y tensión de la tabla 5.1. Se verifica que la fracción volumétrica de hueso sea la misma al principio y al final de la simulación para asegurar el equilibrio. Si la fbm final es menor que la inicial, se aumenta la tensión, y si es mayor, se reduce. Estas condiciones iniciales ajustadas se mantendrán constantes en el proceso de obtención de las densidades para rellenar la matriz 3D. Finalmente, tras el ajuste la tensión tiene un valor de -0.277464 MPa, que se incluye en el modelo junto con la fracción volumétrica con valor 21,84 %.

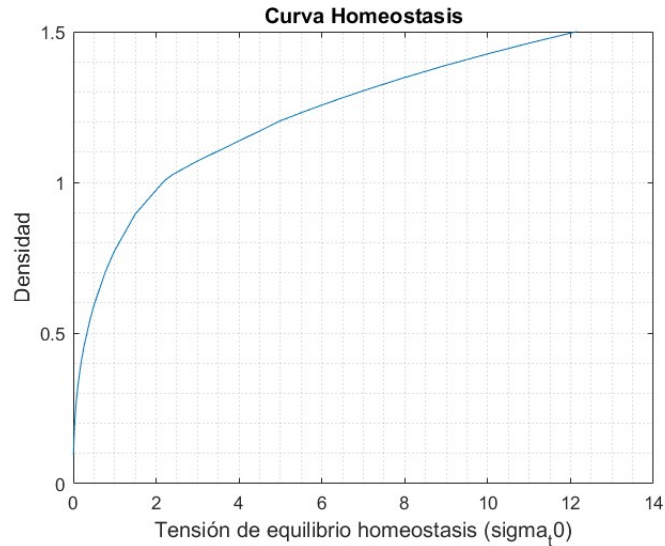


Figura 5.8 Curva de homeostasis densidad.

PASO 2:

Para poder adaptar el modelo matemático se realizan modificaciones muy parecidas a las realizadas para implementarlo con App Designer. Se convierte el Main en una función cuyas entradas son: la fracción volumétrica inicial, la edad, el YSM y el k_5 . Con estas entradas la función devuelve el valor de la densidad para cada valor correspondiente:

- *Main_bone_model_SclAb(fbm_t01,k5,YSM,edad);*

PASO 3:

Se crea un archivo de Matlab para la realización de las diferentes simulaciones. Para ello lo primero es inicializar la matriz con las dimensiones totales. Esas dimensiones serán las siguientes: m3D (20x35x31), ver figura 5.9.

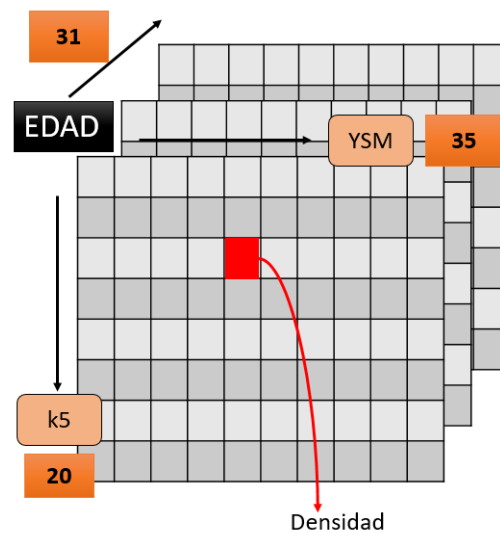


Figura 5.9 Dimensiones de la matriz 3D (Base de datos).

A continuación, se crean vectores que contienen los diferentes valores posibles de dos de los parámetros principales:

- **YSM:** Se define un vector con valores desde 1 hasta 35. El límite superior de 35 se debe a que la última edad de las simulaciones es 80 años, y dado que hemos establecido la edad mínima para la menopausia en 45 años, esto significa que han pasado 35 años desde ese punto, lo que determina el número máximo de años de YSM.
- k_5 Se define un vector con 20 valores desde 0 hasta 0.17 equiespaciados. El valor 0.17 límite de k_5 no es aleatorio. Se han realizado pruebas hasta obtener un valor máximo. El límite de k_5 es el valor más alto de k_5 , para el que se pierde todo el hueso a los 50 años de edad. Es decir, es el punto en el que se observa una pérdida ósea extrema. Este límite es el umbral superior del parámetro k_5 ya que no se espera que ninguna mujer pierda todo el hueso a esa edad.

Una vez definidos los diferentes valores y la matriz que va a ser rellenada, se procede a realizar un bucle para rellenar las columnas de las diferentes hojas de datos. Hay que tener en cuenta que no se rellenan todas las columnas de la matriz en todas las hojas de datos. En el caso de la primera hoja de datos solo se rellenan las 5 primera columnas, esto es debido a que se ha fijado la edad más temprana para la menopausia en 45 años, por lo que si la edad actual es 50 años solo puede haber cinco años desde que comenzó. Por lo tanto, los diferentes bucles cada vez recorren más columnas al ir aumentando el número de valores de YSM al aumentar la edad.

Las simulaciones se realizan por cada hoja de datos. Una vez rellenas todas las hojas de obtiene una base de datos en función de la densidad.

PASO 4: Finalmente, una vez se obtiene la matriz 3D con los datos de las densidades para cada relación de parámetros (edad, k_5 y YSM), esta debe transformarse en una matriz rellena de valores de T-score, que es la variable cuyo valor va a ser conocido gracias al formulario.

Para realizar esta transformación se ha creado un función de Matlab que proporciona dicha matriz 3D final. En este se transforma la matriz 3D de densidades a la matriz 3D de T-score mediante la ecuación 5.1.

5.5.4 Valor k_5 : obtención e implementación en la interfaz

Obtención del valor k_5

Una vez obtenida la matriz 3D de T-score, se desarrolla a continuación el proceso para la obtención del valor de k_5 a partir del T-score introducido en el formulario. Este proceso implica encontrar la hoja de datos correspondiente a la edad y la columna correspondiente al YSM del paciente en la matriz. Después, se realiza la interpolación del T-score específico dentro de la columna seleccionada para obtener el valor de k_5 buscado.

Con el propósito de extraer el k_5 correspondiente se crea una función de Matlab llamada *valork5*, que obtiene este a partir de los valores de edad, edad de la menopausia y T-score del formulario, como puede verse en la ecuación 5.3.

$$k_5 = F(\text{edad}, \text{edad menopausia}, \text{T-score}) \quad (5.3)$$

En la figura 5.10 se puede ver la ruta que utiliza la función de Matlab (herramienta para obtener k_5) para calcular k_5 a partir del T-score. Se parte del valor de T-score del formulario y busca en la matriz 3D, obtenida en el punto 5.5.3, los dos valores inmediatamente más próximos (menor y mayor) al T-score dado. A partir de ellos, interpola el valor de k_5 . Con ese fin, como muestra la figura 5.11, necesita conocer la edad y el YSM del paciente para encontrar la hoja y columna concreta de la matriz 3D.

Finalmente, se obtiene el valor de k_5 deseado para ser implementado en el modelo matemático.

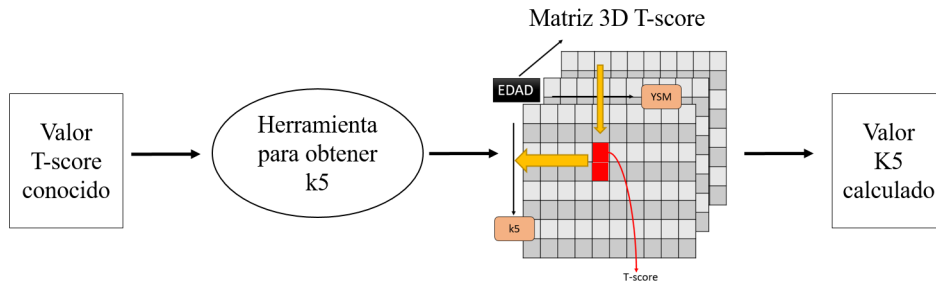


Figura 5.10 Proceso de obtención de k_5 a partir de un T-score dado.

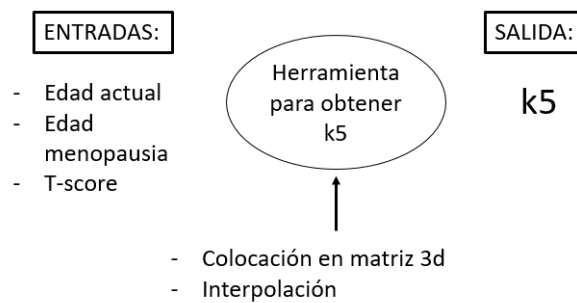


Figura 5.11 Entradas y salidas de la función *valork5*.

Implementación del k_5 en la interfaz gráfica

En la implementación del código de la aplicación en App Designer, se ha incorporado la herramienta de cálculo de k_5 como parte de la ejecución del modelo matemático. En total, se han incluido tres funciones principales en el código de App Designer para lograr el funcionamiento completo del modelo (ver figura 5.12).

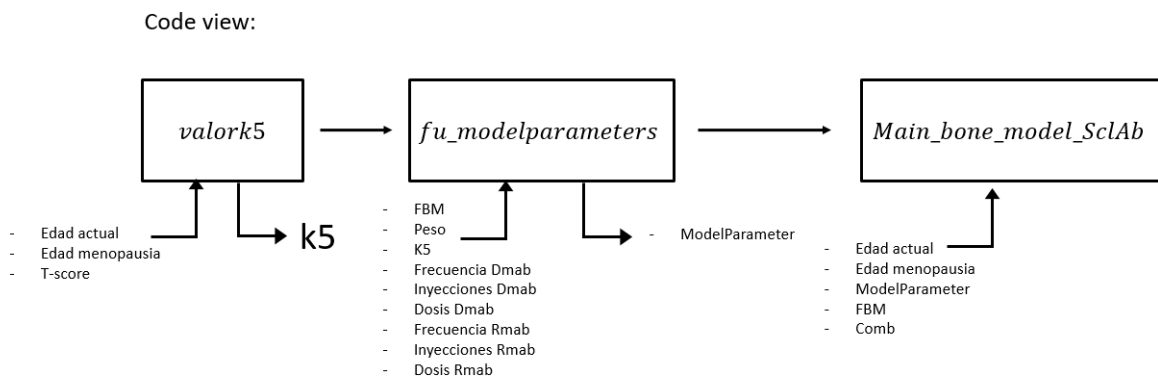


Figura 5.12 Proceso para implementar el k_5 en la interfaz gráfica .

Las funciones, *Main_bone_model_SclAb* y *fu_modelparameters*, han sido previamente explicadas en el punto 5.4.2 . La función *Main_bone_model_SclAb* actúa como la función principal que coordina el flujo de ejecución y controla la interacción con la interfaz de usuario. Por otro lado, la función *fu_modelparameters* se encarga de obtener los datos de entrada necesarios que vienen del formulario

de la aplicación.

La tercera función es *valork5*, cuyo funcionamiento ya se ha explicado en el punto 5.5.4. Esta función, como su nombre indica, se dedica al cálculo del valor de k_5 en función del T-score ingresado por el paciente. Su implementación se basa en la utilización de la matriz 3D de T-score y los parámetros proporcionados.

Por lo tanto, la ejecución del modelo en App Designer se resume:

1. Se llama a la función *valork5* para el cálculo de k_5 .
2. Se llama a la función *fu_modelparameters* para recopilar parte de los datos introducidos en el formulario y el valor de k_5 de la función *valork5*.
3. Se llama a la función *Main_bone_model_SclAb* para ejecutar el modelo y obtener las soluciones solicitadas.

En conclusión, gracias a la implementación de este proceso usando las tres funciones comentadas se logra ejecutar el modelo matemático de manera sencilla. Los usuarios pueden obtener los resultados deseados directamente desde la interfaz, sin la necesidad de acceder al código del modelo ni tener conocimientos previos sobre su funcionamiento interno.

5.6 Implementación de la interfaz

Tras conocer el funcionamiento interno y el diseño de la calculadora, se procede a exponer como funciona. Se realiza un ejemplo con los datos concretos de la tabla 5.2.

Tabla 5.2 Parámetros del ejemplo de la interfaz calculadora.

Parámetro	Valor con Unidades
Edad Actual	61 años
Edad de la Menopausia	50 años
Peso	60 kg
T-score	2.6
Dosis de Denosumab	70 mg
Frecuencia de Denosumab	150 días
Número de Inyecciones	10 inyecciones

Se muestra, al principio de este capítulo, en la figura 5.1 la interfaz completa.

Para utilizar la interfaz, se procede primero a completar la sección correspondiente del formulario ("Questionnaire", en la figura 5.1). Una vez que se haya completado con los datos correspondientes, se debe presionar el botón "Terminar" ("End" en la figura 5.1). Después de presionar dicho botón, las evoluciones de la densidad ósea, fracción volumétrica de hueso y niveles de daño se muestran por pantalla.

Tras esta primera parte, se rellena la segunda para evaluar la probabilidad de fractura. Una vez rellena, se pulsa el botón PF (ver figura 5.1) y aparece en pantalla la evolución de la probabilidad de fractura. Se puede ver en la tabla resumen 5.3 los campos a rellenar en ambas parte de la interfaz y las salidas que dan lugar tras pulsar los botones.

Se muestra en la figura 5.13 la interfaz rellena y en las figuras 5.14 y 5.15 las evoluciones de densidad, fracción volumétrica y daño para el caso concreto.

Finalmente, se muestra en la figura 5.16 la parte 2 de la interfaz rellena y en la figura 5.17 dos evoluciones de probabilidad de fractura concretas. Una de ellas para la vertebra a tracción y otra para la vertebra a compresión. Se pueden modificar los datos de la pronosticadora de PF y pulsando de nuevo el botón aparece la nueva evolución.

Tabla 5.3 Entradas y salidas de las secciones de la interfaz.

PARTE 1: FORMULARIO	
Entradas	Salidas
Información Inicial	Evolución de la Densidad Aparente
Información del Tratamiento	Evolución de la Fracción Volumétrica de Hueso
	Evolución del Daño
PARTE 2: PRONOSTICADOR DE PF	
Entradas	Salida
Archivo de datos de la Parte 1	Probabilidad de Rotura
Sobrecarga	
Hueso Concreto	
Tracción o Compresión	

QUESTIONNAIRE

GENERAL PATIENT'S DATA

Name Date of Birth
Age Year
Sex Month
Age at menopause Day

INFORMATION PROVIDED TO FRAX

Height (cm) Alcohol 3 or more units/day (Y/N)
Previous Fracture (Y/N) Current Smoking (Y/N)
Parent Fractured Hip (Y/N) Rheumatoid arthritis (Y/N)
Weight (kg) Glucocorticoids (Y/N)
Secondary osteoporosis (Y/N)

INFORMATION OF THE DRUG TREATMENT

Drug <input checked="" type="checkbox"/> Dmab <input type="checkbox"/> Rmab <input type="checkbox"/> Rmab+Dmab	Anatomical site <input checked="" type="checkbox"/> Vertebra <input type="checkbox"/> Femur	Treatment data Dmab Dose Dmab (mg) <input type="text" value="70"/> Time between 2 inj Dmab (days) <input type="text" value="150"/> # injections Dmab <input type="text" value="10"/>
P_RANKL_d_PMO <input type="text" value="0"/> T-score <input type="text" value="2.6"/>		Treatment data Rmab Dose Ramb (mg) <input type="text" value="0"/> Time between 2 inj Rmab (days) <input type="text" value="0"/> # injections Rmab <input type="text" value="0"/>

Figura 5.13 Parte 1 de la interfaz rellena.

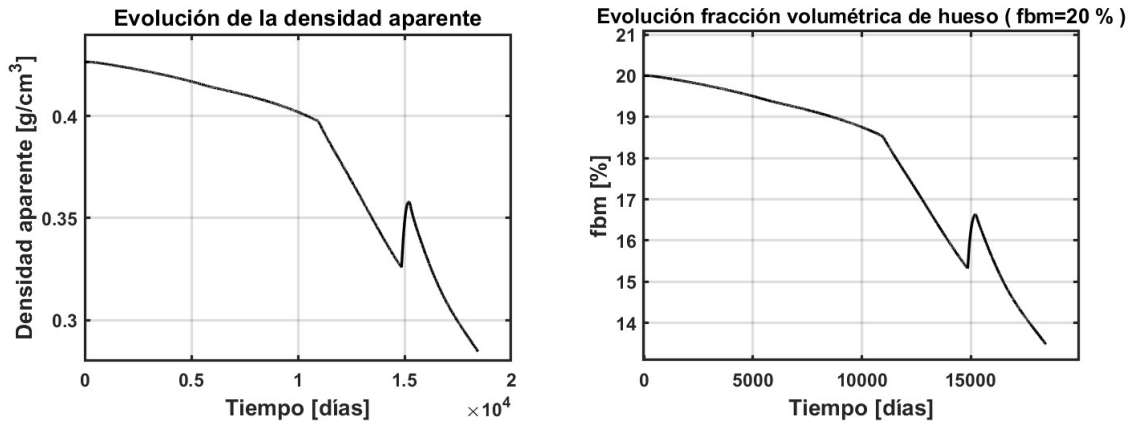


Figura 5.14 Evolución de la densidad y fracción volumétrica.

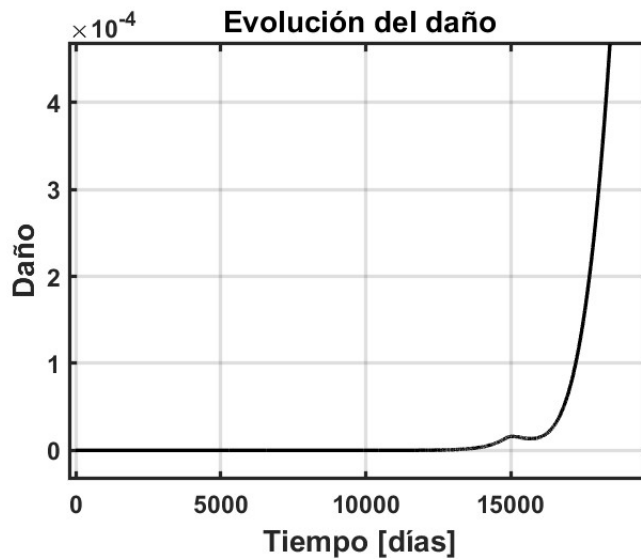


Figura 5.15 Evolución del daño.

FRACTURE RISK (FR) PREDICTION

Name

Overload

Anatomical site <input checked="" type="checkbox"/> Vertebra <input type="checkbox"/> Femur	Type of load (vertebra) <input checked="" type="checkbox"/> Tension <input type="checkbox"/> Compression	Type of load (femur) <input type="checkbox"/> Tension <input type="checkbox"/> Compression
---	--	--

Figura 5.16 Parte 2 de la interfaz rellena.

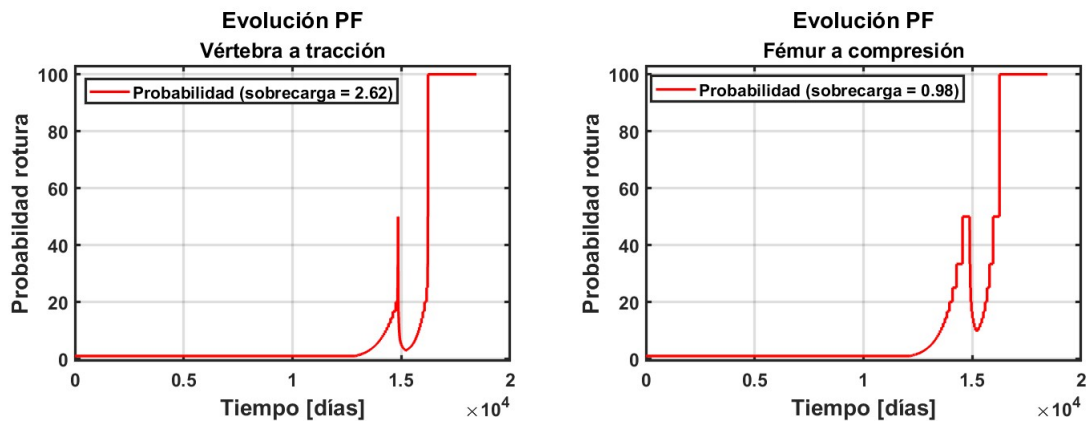


Figura 5.17 Probabilidad de fractura concreta para la vertebra a tracción y probabilidad de fractura concreta para el fémur a compresión.

6 Redes Neuronales

En este capítulo, se abordan las bases de las redes neuronales, sus clasificaciones principales y cómo crearlas en Matlab. Además, se presentará la Toolbox de Aprendizaje Profundo de Matlab para diseñar y entrenar redes neuronales.

6.1 Introducción a las redes neuronales

6.1.1 Definición de redes neuronales

Machine learning

Es una disciplina del campo de la Inteligencia Artificial (IA), que a través de algoritmos, dota a los ordenadores de la capacidad de identificar patrones en datos masivos y elaborar predicciones. Esto permite a los ordenadores realizar un análisis predictivo [76].

Redes Neuronales

Una red neuronal es un método de IA que enseña a las computadoras a procesar datos de una manera que está inspirada en la forma en la que lo hace el ser humano. Es un tipo de Machine Learning que utiliza neuronas interconectadas en una estructura de capas que se asemeja al cerebro humano. Son por tanto un conjunto de algoritmos diseñados especialmente para reconocer patrones [77].

6.2 Redes neuronales artificiales

6.2.1 Neurona biológica

La neurona es una célula del sistema nervioso que posee la capacidad de recibir y decodificar información en forma de señales eléctricas y químicas, transmitiéndolas a otras células. Cada neurona se compone de tres partes: el cuerpo o soma que es la parte central de la neurona donde se encuentra el núcleo y orgánulos celulares. El axón que es la estructura larga y delgada que se extiende desde el cuerpo de la célula y es el responsable de llevar los impulsos eléctricos generados en el cuerpo hasta las terminaciones nerviosas. Finalmente, se encuentran las dendritas que son estructuras ramificadas que se extienden desde el cuerpo de la célula. Estas reciben los impulsos eléctricos y químicos de otras neuronas o células, transmitiendo la información al cuerpo celular.

6.2.2 Neuronas artificiales

Una neurona artificial es un procesador elemental, analizando un vector de entrada y produciendo una respuesta o salida única. También se denomina perceptrón. Los principales elementos de una neurona artificial se pueden observar en la figura 6.1 y son los siguientes [78]:

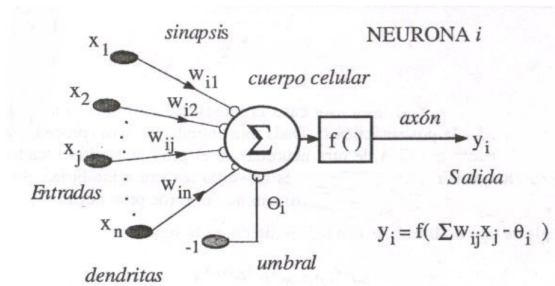


Figura 6.1 Neurona artificial. Tomada de [78].

- Las entradas x_{ij} , reciben los datos de otras neuronas que equivalen a las dendritas en el caso de las neuronas biológicas.
- Los pesos sinápticos w_{ij} . En una neurona artificial los pesos se asignan a las conexiones entre las entradas de una neurona y el axón de otra. Estos pesos representan la importancia relativa de cada conexión y se ajustan durante el entrenamiento de la red neuronal (se explicará más adelante). El peso por lo tanto almacena la información de la red.
- Una regla de propagación $h_i(t)$. Utilizando las entradas y los pesos sinápticos se suele realizar una operación para obtener el valor del potencial postsináptico. La suma ponderada suele ser la operación más común dando mayor importancia a ciertas entradas según su peso (ver figura 6.2).

$$h_i(t) = \sum_j w_{ij}x_j$$

Figura 6.2 Regla más común de propagación. Tomada de [78].

- Una función de activación. El valor obtenido con la regla de propagación, se filtra a través de una función conocida como función de activación y es la que proporciona la salida de la neurona. Existen varios tipos de función de activación y su elección va en función de la utilidad que se le quiera dar a la red neuronal. En la siguiente figura 6.3 se muestran las funciones de activación más habituales.
- El umbral o bias es un valor ajustable que se utiliza para determinar si la neurona se activa o no en función de la suma ponderada de las entradas. Indica a partir de que valor del potencial postsináptico la neurona produce una salida significativa.

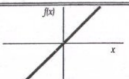
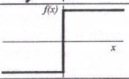
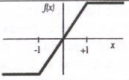
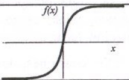
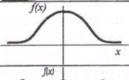

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq l \\ +1, & \text{si } x > l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(ax + \varphi)$	$[-1, +1]$	

Figura 6.3 Diferentes funciones de propagación. Tomada de [78].

6.3 Arquitectura de una red neuronal

6.3.1 Capa neuronal

Las unidades de procesamiento se organizan en capas, ver figura 6.4. Se conoce como capa neuronal a la agrupación de un conjunto de perceptrones que trabajan a la vez. Las redes neuronales están normalmente compuestas por tres partes: una capa de entrada, una o varias capas ocultas y una capa de salida. Los datos de entrada se presentan en la primera capa y los valores se propagan desde cada neurona de esta hasta cada neurona de la capa siguiente, al final se obtiene un resultado que es enviado desde la capa de salida. A mayor número de capas ocultas se dice que la red neuronal es más profunda.

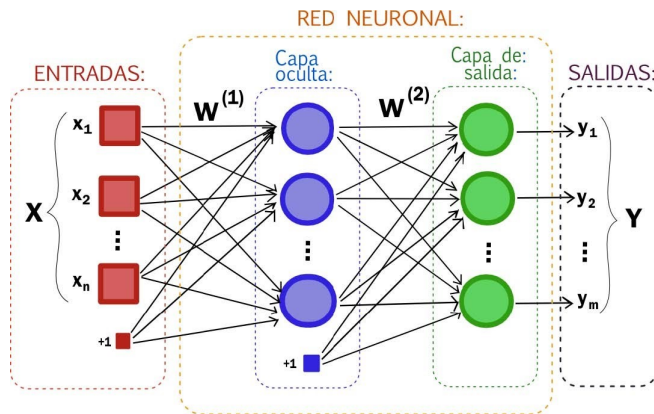


Figura 6.4 Capas de una red neuronal. Tomada de [79].

6.3.2 Clasificación de las redes neuronales

Se pueden clasificar las redes neuronales en función de los siguientes parámetros [80]:

Clasificación por el número de capas:

- **Redes Neuronales monocapa:** Solo se contabiliza la capa de salida al ser en la que se realizan los cálculos. La capa de entrada se conecta directamente a la de salida, ver figura 6.5.

- **Redes Neuronales multicapa:** Entre las capas de entrada y salida existen, existen diversas capas que se denominan capas ocultas y que pueden conectarse entre ellas o no.

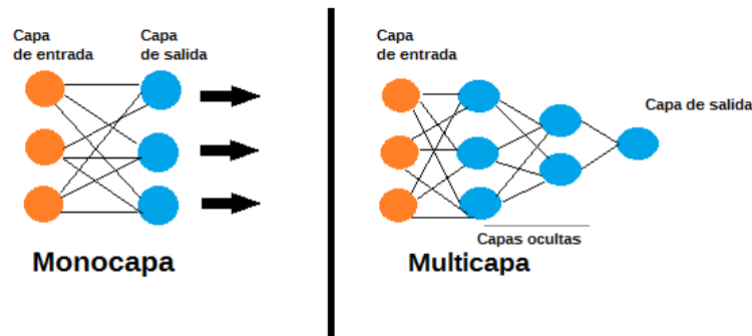


Figura 6.5 Monocapa vs multicapa.

Clasificación por los tipos de conexión:

- **Redes neuronales no recurrentes:** La información de la red trabaja en un solo sentido. No existe una realimentación y carecen de memoria.
- **Redes neuronales recurrentes:** Las neuronas pueden tener conexiones de realimentación ya sea entre neuronas de la misma capa o de diferentes, ver figura 6.6. Esta realimentación permite la memoria en este tipo de redes. Suelen ser más potentes que las no recurrentes.

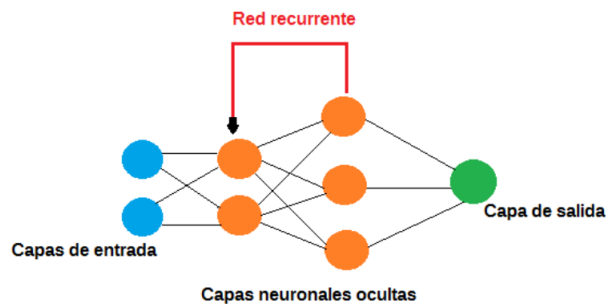


Figura 6.6 Red neuronal recurrente.

Clasificación por grados de conexión:

- **Redes neuronales totalmente conectadas:** Las capas anteriores y posteriores están completamente conectadas teniendo la totalidad de las neuronas conectadas entre ellas.
- **Redes parcialmente conectas:** No todas las neuronas están conectadas.

6.3.3 Tipos de redes neuronales

Existen diferentes tipos de redes neuronales, pero cabe destacar tres de ellos por encima del resto [81]:

- **Redes Neuronales Convolucionales (CNN):** este tipo de red utiliza la operación de convolución como base para el procesamiento de los datos. Son un tipo de arquitectura de redes neuronales especializadas en el procesamiento de datos estructurados en forma de mallas o matrices. Cada neurona no se une con todas y cada una de las capas siguientes, ver figura 6.7, sino que solo con un subgrupo de ellas (se especializa), con esto se consigue reducir el número de neuronas necesarias y la complejidad computacional necesaria para su ejecución.

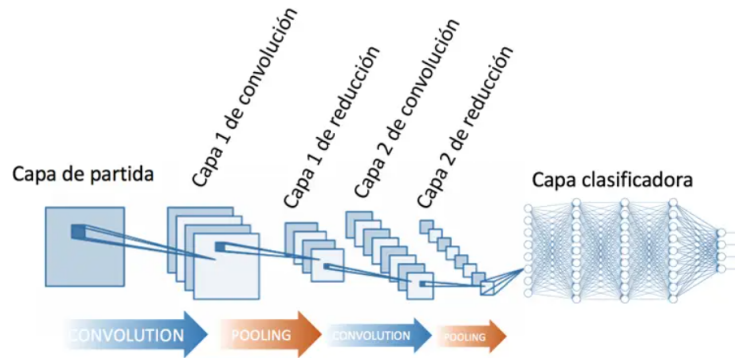


Figura 6.7 Redes Neuronales Convolucionales. Tomada de [81].

- **Redes Neuronales Recurrentes (RNN):** Son aquellas capaces de influir sobre si mismas a la hora de entrenarse, ver figura 6.8. Dada una RNN, cada neurona de la misma devuelve dos parámetros, el habitual output o coste y otro que representa la información obtenida en ella, que a su vez es traspasada a la siguiente neurona para influir sobre ella y así crear la recurrencia.

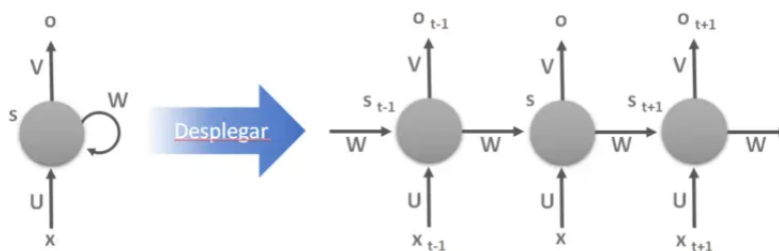


Figura 6.8 Redes Neuronales Recurrentes. Tomada de [81].

- **Redes Neuronales densas:** Son aquellas en las que todas las neuronas de una capa están conectadas a todas las de la siguiente capa, ver figura 6.9. Estas redes, son más eficaces en cuanto al tratamiento de información porque cada neurona dispone de más datos para tratar, pero a su vez, son más lentas a la hora de procesar los datos debido la gran cantidad de información que mueven de una capa a otra.

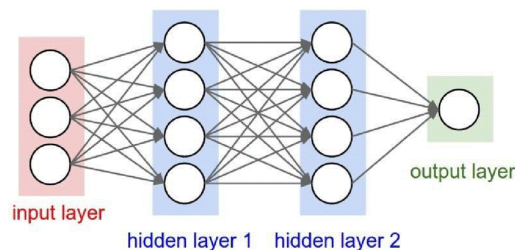


Figura 6.9 Redes Neuronales densas. Tomada de [82].

- **Redes Neuronales feedforward:** También conocida como red neuronal de propagación hacia delante, en la que la información solo fluye hacia un sentido, desde la capa de entrada a la capa de salida, ver figura 6.10. Es un tipo de red neuronal artificial que se utiliza en la mayoría

de las aplicaciones de aprendizaje automático y reconocimiento de patrones [83]. La red neuronal feedforward consta de múltiples capas:

- Una capa de entrada
- Una o más capas ocultas
- Una capa de salida

Cada capa consta de un conjunto de nodos o neuronas, que reciben entradas de la capa anterior y generan salidas para la capa siguiente. Los pesos entre los nodos se ajustan durante el entrenamiento de la red neuronal para minimizar el error en la predicción de la salida [83]. El proceso de predicción en una red neuronal feedforward implica alimentar los datos de entrada a través de la capa de entrada y las capas ocultas y, finalmente producir una salida en la capa de salida. Cada capa de la red neuronal utiliza una función de activación no lineal para transformar las entradas y generar las salidas correspondientes. Las funciones de activación más comunes son la función sigmoide y la función ReLU [83]. Las redes neuronales feedforward son utilizadas en una amplia variedad de aplicaciones de aprendizaje automático, incluyendo la clasificación de imágenes, el procesamiento del lenguaje natural, el reconocimiento de voz, entre otros.

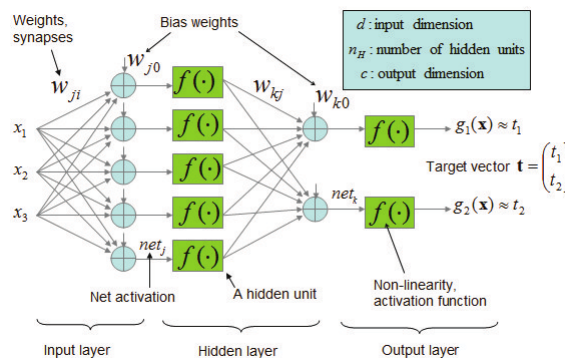


Figura 6.10 Red Neuronal feedforward. Tomada de [84].

6.4 Métodos de aprendizaje

Una de las características más importantes de las redes neuronales es su poder de aprendizaje. Este puede definirse como la modificación del comportamiento de la red resultante de la aplicación sucesiva de estímulos. Esta modificación tiene lugar mediante la variación de los pesos sinápticos de las neuronas [85]. Dependiendo de su poder de aprendizaje esos algoritmos se clasifican en tres grupos [86]:

- Aprendizaje supervisado: Son aquellos modelos que aprenden en base a unas entradas y unas salidas conocidas de antemano.
- Aprendizaje no supervisado: Este tipo de aprendizaje se basa en que solo se disponen de los datos de entrada y no de salida, ya que el objetivo del estudio es conocer estructuralmente los datos disponibles con el fin de encontrar patrones entre ellos para ordenarlos de alguna manera.
- Aprendizaje reforzado: En este método el algoritmo aprende en base a prueba y error mejorando la respuesta en base a un algoritmo de retroalimentación.

6.5 Funcionamiento de una red neuronal

La red neuronal consiste en un conjunto de interconexiones de unidades llamadas neuronas artificiales organizadas en capas. La información de la red neuronal se procesa en etapas. Estas utilizan elementos simples que operan en paralelo, y su funcionamiento está inspirado en el de sistemas nerviosos biológicos. Las redes reciben una serie de valores de entrada y cada una de ellas llega a una neurona. Cada una de las neuronas posee un peso, que es un valor numérico, con el que modifica la entrada recibida. Estos nuevos valores obtenidos salen de la neurona correspondiente para continuar su camino por la red (ver figura 6.11). El final de la red es una salida que es la predicción calculada.

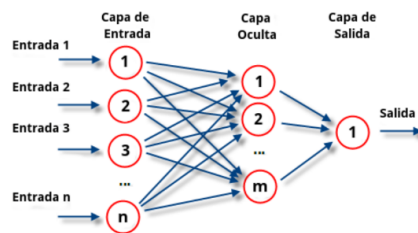


Figura 6.11 Red neuronal. Tomada de ..

Para conseguir el funcionamiento de una red es necesario entrenarla. Este entrenamiento consiste en la modificación de los pesos para conseguir el resultado deseado. Para ello lo que se hace es introducir datos de entrenamiento en la red, en función del resultado que se obtenga, se modificarán los pesos de las neuronas en función del error obtenido y en función de cuanto haya contribuido cada neurona. Esto se conoce como el backpropagation o propagación hacia atrás [83]. Este algoritmo de retro propagación consiste en propagar el error desde la capa de salida hacia las capas ocultas, ajustando los pesos de las conexiones en función de la contribución de cada neurona al error total. Este proceso permite que la red "aprenda" a través de múltiples iteraciones y ajuste sus pesos para mejorar su capacidad de hacer predicciones precisas.

Este proceso consta de varias fases [83]:

1. **Inicialización de los pesos:** Se inicializan de forma aleatoria o mediante alguna técnica específica.
2. **Paso hacia delante (forward propagation):** Se introducen los datos de entrenamiento en la red neuronal y se propagan hacia delante desde la capa de entrada hasta la capa de salida. Cada neurona realiza la suma ponderada de las salidas de las neuronas de la capa anterior, aplica una función de activación y pasa el resultado a las neuronas de la capa siguiente.
3. **Cálculo de la función de pérdidas:** Una vez se obtiene la salida, se calcula la diferencia entre esta y los valores deseados para el conjunto de entrenamiento. Esto se realiza utilizando una función de pérdida que cuantifica como de lejos están los valores de salida de los deseados.
4. **Retropropagación del error:** El error se propaga desde la capa de salida hacia la capa de entrada. Se calcula el gradiente de la función de pérdida con respecto a los pesos de las conexiones entre las neuronas. Esto se hace aplicando la regla de la cadena para calcular las derivadas parciales del error en relación con cada peso.
5. **Actualización de pesos:** Utilizando el gradiente calculado en el paso anterior, se ajustan los pesos de las conexiones mediante un algoritmo de optimización como el gradiente del descenso

estocástico. El objetivo de esto es minimizar la función de pérdidas, ajustando los pesos en la dirección que se reduce el error.

- 6. Repetición del proceso:** Este proceso se realiza para todos los datos de entrenamiento y aunque puede realizarse solo en una pasada se suele hacer en varias, denominando cada número de pasadas del conjunto total de los datos de entrenamiento: epoch. El número de epoch a realizar (recorrido completos a todos los datos de entrenamiento) dependen del problema y la convergencia del modelo. Si el número de epoch es demasiado bajo puede que la red no pueda alcanzar un rendimiento alto mientras que si el número es demasiado alto existe el riesgo de sobrentrenar la red haciendo que esta no se capaz de generalizar nuevos datos que no sean del conjunto de entrenamiento.

6.6 Implementación de redes neuronales en Matlab

Herramientas y paquetes de trabajo

Matlab ofrece toolboxes especializadas para trabajar con Machine Learning, Redes Neuronales, Deep Learning, Visión Artificial y Conducción Autónoma [87]. Utilizando solo unas pocas líneas de código Matlab te permite desarrollar Redes Neuronales sin tener una gran experiencia. En este trabajo se ha usado concretamente la ToolBox de Matlab llamada, 'Deep Learning Toolbox'.

Creación, entrenamiento y evaluación de una red neuronal

El flujo típico de trabajo para diseñar una red neuronal es el siguiente [87]:

1. Preparar los datos necesarios y el acceso a ellos.
2. Crear la red neuronal.
3. Configurar las entradas y salidas de la red.
4. Ajustar los parámetros de la red para optimizar el rendimiento.
5. Entrenar la red.
6. Validar los resultados de la red.
7. Integrar la red.

Teniendo como base el índice anterior se plasma de una manera más detallada. En primer lugar, se deben definir los datos de entrenamiento: los de entrada (parámetros o características) y los de salida (objetivos). Este conjunto se utilizará para entrenar la red. En el caso de estudio estos datos se obtendrán en función de un modelo matemático de Remodelación Ósea. A continuación, es necesario crear la red neuronal a partir de la función `feedforwardnet`. Esta función permite crear una red neuronal compuesta por una capa de entrada, una o varias capas ocultas y una capa de salida. La primera entrada es el número de capas ocultas y su respectivo número de neuronas. La segunda entrada es la función de entrenamiento, existen tres:

- `Trainlm` (Levenberg Marquardt): [87]
 - Es un algoritmo de entrenamiento iterativo que utiliza métodos de optimización no lineales.
 - Es adecuado para problemas de regresión y clasificación.
 - Es conocido por ser rápido y eficiente en términos de convergencia.
 - Requiere más memoria que algunos otros algoritmos de entrenamiento.

- Trainbr (Backpropagation con regulación de Bayes): [87]
 - Es un algoritmo de entrenamiento que combina el algoritmo de retropropagación clásico con la regulación de Bayes.
 - La regulación de Bayes ayuda a evitar el sobreajuste (overfitting).
 - Funciona bien con un conjunto de datos pequeños. Mientras que con conjuntos de datos más grandes puede ser más lento en comparación con otros.
- Traincg (Conjugate Gradient Backpropagation) : [87]
 - Es un algoritmo de entrenamiento basado en el método del gradiente conjugado.
 - Combina el algoritmo de retropropagación con técnicas de optimización basadas en la conjugación de gradientes.
 - Suele ser más rápido que que trainlm en conjuntos grandes, mientras que en conjuntos pequeños es menos preciso.

La elección de la función de entrenamiento depende de las características del problema, la cantidad de datos y los resultados deseados. Experimentar con los algoritmos es una buena forma de encontrar la respuesta que se adapte a las necesidades específicas.

```

1 Código Matlab
2
3 hiddenSize = 10; % Número de neuronas en la capa oculta
4 trainFcn = trainbr; % trainbr/trainlm/traincg % Función de entrenamiento (3
   opciones)
5 net = feedforwardnet(hiddenSize,trainFcn); % Crear una red neuronal feedforward

```

Una vez definida la arquitectura de la red, debe configurarse. Esto se realiza con la función *configure* de Matlab. La información necesaria para esta función son la red creada anteriormente (net) y los datos de entrenamiento; entrada y salida:

```

1 Código Matlab
2
3 % x': Datos de entrada (parámetros)
4 % y': Datos de salida (objetivo)
5
6 net = configure(net, x', y'); % Configurar la red con los datos de entrada y
   salida

```

Una vez configurada, ya esta lista para el entrenamiento. Para realizarlo se utiliza la función de Matlab *train*. Esta función necesita como entrada la red, y los datos de entrenamiento. Las salidas son dos:

- **'net'**: La red neuronal entrenada. Esta contiene los datos de los pesos y sesgos finales tras el entrenamiento. Esta red es la que se usará para predecir soluciones en función de datos nuevos.
- **'tr'**: Es una estructura que contiene información sobre el proceso de entrenamiento. Entre estos datos se encuentra el número de epoch, el rendimiento de la red en cada uno de los epoch, datos sobre la validación, etc. Algunos ejemplos de información proporcionada por tr:
 - tr.trainInd: Índices de los datos utilizados para el entrenamiento.

- tr.epoch: Número de epoch (iteraciones) realizadas durante el entrenamiento.
- tr.perf: Rendimiento de la red neuronal en cada epoch.

```

1 Código Matlab
2
3 [net,tr] = train(net, x', y'); % Entrena la red con los datos de entrada y
   salida
4
5 view(net)

```

Finalmente, una vez se tiene la red neuronal entrenada se puede guardar usando el comando `save` de Matlab. Para poder usar la red creada posteriormente se guarda como una función a la que le entran los parámetros (datos de entrada) para los que se quieren predecir una solución y como salida se tiene esta predicción.

```

1 Código Matlab
2
3 save net % Se guarda de red entrenada
4 genFunction(net, 'funcionRN') % Creación de un función para predicciones
   posteriores

```

6.7 Datos necesarios

Como se ha comentado anteriormente, para el entrenamiento de la red neuronal son necesarios los datos de entrada y los datos de salida. Estos datos vendrán dados en forma de matrices. Estos conjuntos de datos se obtienen del modelo matemático de remodelación ósea explicado en el capítulo 3. Los parámetros de entrada son los mismos que necesita el modelo para el cálculo de las soluciones.

Matriz de entrada a la red

- **Filas:** Representan los diferentes casos de estudio
- **Columnas:** Representan los parámetros de entrada al modelo..

Matriz de salida de la red

- **Filas:** Representan los diferentes casos de estudio.
- **Columnas:** Representan los diferentes instantes de tiempo donde de evalúan los parámetros de salida.

En Matlab, es común dividir los datos de entrada en tres subconjuntos distintos: entrenamiento, validación y testeo. Esta división se realiza para poder evaluar el rendimiento de la red neuronal y evitar el sobreajuste [87]:

- El **conjunto de entrenamiento** se utiliza para entrenar la red. Su propósito es ajustar los pesos y los sesgos de la red neuronal. Esta se alimenta con ejemplos de entrada y salida conocidos. Procesando estos datos ajusta los pesos mediante algoritmos de optimización buscando minimizar la diferencia entre la salida predicha y la real. El objetivo del conjunto es proporcionar los datos necesarios para entrenar y ajustar los parámetros internos de la red mejorando así gradualmente su rendimiento.

- El **conjunto de validación** se utiliza para evaluar el rendimiento de la red durante el proceso de entrenamiento y así poder realizar ajustes en los parámetros o arquitectura de la red. Este conjunto permite evaluar el rendimiento del modelo en datos no vistos durante el entrenamiento y tomar decisiones sobre los ajustes necesarios para mejorar el rendimiento. Por lo tanto, el objetivo de este subconjunto es seleccionar la mejor configuración del modelo.
- El **conjunto de prueba o testeo** se utiliza para evaluar la capacidad de generalización de la red después del entrenamiento midiendo el rendimiento final de esta una vez completado el entrenamiento del modelo y seleccionando una configuración óptima se evalúa el rendimiento de este con unos datos nuevos proporcionando así una visión realista de la capacidad del modelo para generalizar y realizar predicciones. Su objetivo es proporcionar una medida del rendimiento final de la red.

La separación de los datos puede variar según la aplicación. La proporción más típica y propuesta por Matlab es 70% para entrenamiento, 15% para validación y 15% para prueba. Sin embargo, estos valores se pueden ajustar si fuera necesario.

7 Interfaz optimizadora

En el presente capítulo, se lleva a cabo el desarrollo de la creación de la interfaz denominada "Optimizador" en la plataforma App Designer de Matlab. Este proceso comprende la creación de la interfaz utilizando las herramientas proporcionadas por Matlab, así como su posterior integración en App Designer.

7.1 Optimización

La optimización se encarga de encontrar la mejor solución posible para un problema, generalmente expresado como una función objetivo sujeta a ciertas restricciones. El objetivo es encontrar los valores de las variables de decisión que minimicen o maximicen dicha función objetivo. Un algoritmo de optimización es un conjunto de reglas diseñadas para encontrar la mejor solución a la función objetivo, la solución óptima. Se utiliza un algoritmo de optimización en este trabajo para calcular el tratamiento óptimo de denosumab, partiendo de las redes neuronales descritas en el capítulo 6.

7.1.1 Algoritmos de optimización

Un algoritmo de optimización se puede definir como una transformación de un problema de optimización en una función de probabilidad, que utiliza un conjunto de operaciones de búsqueda iterativas para encontrar al menos un óptimo local.

Dentro de los algoritmos de optimización existen dos tipos: los locales y los globales. Los algoritmos de optimización local tratan de acceder a los extremos de una función dentro de una región, la solución que se encuentra no tiene por que ser la absoluta. Mientras que, la optimización global, no se limita solo a una posible solución, implica encontrar la solución óptima a problemas que contienen óptimos locales.

Los optimizadores locales se pueden clasificar en dos categorías:

- **Deterministas:** No tiene componentes estocásticos y garantizan la convergencia al óptimo global siempre y cuando la función objetivo cumpla con ciertas propiedades.
- **Estocásticos:** Estos algoritmos evalúan la función en una muestra de puntos elegidos al azar y continúan iterativamente evaluando la función en otros puntos. No se puede garantizar la convergencia al óptimo global. Al contrario que los deterministas, este enfoque no requiere que la función objetivo cumpla con una especificaciones.

7.1.2 Optimización en Matlab

El toolbox de optimización de Matlab contiene una colección amplia de funciones que permiten minimizar (o maximizar) funciones no lineales generales. La función para resolver los problemas

de optimización no lineales con restricciones en Matlab es *fmincon*. En el contexto de este trabajo, la optimización que se quiere abordar es la de un tratamiento de denosumab, esto implica minimizar un conjunto de variables (los parámetros de diseño del tratamiento) de manera que se cumplan una serie de especificaciones o características necesarias para el tratamiento (restricciones en dosis y lapso entre inyecciones). Se trata, por lo tanto, de un problema de optimización no lineal con múltiples variables. Para abordar esta optimización, se ha empleado la función de minimización *fmincon*.

Se utiliza en ella, el algoritmo predeterminado por la función denominado Algoritmo de Punto Interior. La idea de este algoritmo es trabajar dentro del espacio factible del problema, sin buscar una solución fuera de las restricciones.

- **Inicialización:** El algoritmo empieza desde un punto inicial factible dentro del espacio restringido.
- **Iteración:** en cada iteración, intenta mejorar la solución actual, moviéndose hacia el mínimo de la función objetivo, siempre dentro de las restricciones. Para hacerlo, utiliza técnicas basadas en la resolución de un sistema de ecuaciones que involucra el gradiente de la función objetivo y las restricciones.
- **Paso de barrera:** El algoritmo utiliza un parámetro denominado barrera que controla la cercanía a las restricciones. Este parámetro va disminuyendo gradualmente hasta alcanzar la solución óptima.
- **Convergencia:** Se repiten las interacciones hasta que se cumplan ciertos criterios de convergencia, como la convergencia del gradiente y la factibilidad de las restricciones.
- **Solución óptima:** Una vez cumplidos los criterios de convergencia se proporciona una solución que es un mínimo local o global según el punto inicial y el problema.

La sintaxis necesaria para este trabajo es la siguiente:

```
[x,fval,existflag,output] =
    fmincon('fun',x0,AI,bI,A,b,lb,ub,'NonConFunc',Options)
```

Parámetros de entrada:

- **fun:** Función escalar objetivo.
- **x0:** Solucion inicial.
- **AI y bI:** Definen las restricciones de desigualdad (si no existen éstas se definen AI=[] y bI=[]).
- **A y b:** Definen las restricciones de igualdad (si no existen éstas usar A=[] y b=[]).
- **lb y ub:** Definen las cotas superiores e inferiores de las incógnitas.
- **Options:** estructura con las opciones de optimización (número de iteraciones, tolerancia, algoritmo, entre otros).

Parámetros de salida:

- **x:** Solución óptima.
- **fval:** valor de la función objetivo en el mínimo.
- **Output:** Es una estructura que proporciona información adicional sobre el proceso de optimización.

7.2 Interfaz optimizadora

El objetivo previamente establecido consiste en desarrollar una interfaz que proporcione un tratamiento óptimo personalizado de denosumab. Para lograrlo se emplean las redes neuronales implementadas en Matlab, tal como se explicó en el capítulo 6, combinadas con la optimización definida en el punto 7.1.2.

El proceso, de manera general (ver figura 7.1, se desarrolla de la siguiente manera: en primer lugar, se entrenan una serie de redes neuronales utilizando Matlab para obtener los parámetros dinámicos que proporciona el modelo matemático explicado en el capítulo 3. La elección de las redes neuronales se basa en la necesidad de reducir significativamente el tiempo computacional, que sería considerablemente extenso si se empleara directamente el modelo matemático.

Una vez completado el entrenamiento de las redes neuronales, se implementa un proceso de optimización en Matlab, utilizando la función *fmincon*, que incorpora penalizaciones en los valores de fracción de ceniza y el daño. Este proceso de optimización incluye las redes neuronales entrenadas. Finalmente, una vez definida la optimización, se lleva a cabo la integración de esta en App Designer, proporcionando así una interfaz gráfica para el cálculo del tratamiento óptimo.

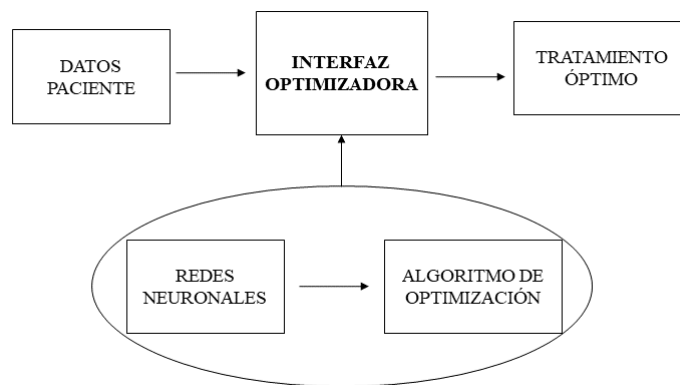


Figura 7.1 Proceso de obtención de la interfaz .

7.2.1 Redes neuronales

El primer paso en la creación de la interfaz es la capacidad para predecir los parámetros dinámicos de la paciente que se necesitan para la elección del tratamiento. Estos parámetros incluyen el daño, la fracción de ceniza, la fracción volumétrica de hueso y la cantidad de osteoclastos activos. Esto se logra mediante el empleo de redes neuronales. La razón detrás de este enfoque es la eliminación de la dependencia del uso del modelo matemático detallado en el capítulo 3, junto con su alta carga computacional.

Para lograrlo es necesario entrenar previamente las redes neuronales con conjuntos de datos que son proporcionados por el modelo matemático. Estos conjuntos de datos sirven como un banco de información con el que la red puede aprender a predecir. Una vez que hayan aprendido a predecir estos parámetros a partir de este amplio conjunto de datos, se pueden obtener los valores de los parámetros para cualquier caso específico sin necesidad de haber sido previamente evaluado.

Obtención de los conjuntos de datos para las redes neuronales

Como se explicó anteriormente en el capítulo 6 dedicado a las redes neuronales, es necesario contar con un conjunto de datos de entrada y salida para entrenar una red neuronal y que así, esta pueda predecir resultados basados en combinaciones de parámetros no estudiados anteriormente. En este caso, para obtener los conjuntos de datos, se parte del modelo matemático desarrollado en el capítulo

3. Este modelo determina la variación de las variables dinámicas del problema que son las salidas que se quieren predecir con la red neuronal. A partir de este modelo y utilizando las variables de la tabla 7.1, las cuales son las entradas de la red neuronal, se pueden determinar las variables de salida. Como se debe contar con un gran número de datos para entrenar la red, se necesitan múltiples combinaciones de los parámetros de entrada. Con estas diferentes combinaciones y el modelo matemático, se obtienen las correspondientes salidas.

A continuación, se describen los siete parámetros de entrada para el entrenamiento:

- **Edad actual de la paciente**
- **Edad de menopausia**
- k_5 : Modela el incremento de producción de RANKL, teniendo así en cuenta en el modelo el efecto de la menopausia y, con ello, la gravedad de la osteoporosis.
- **Peso**: Peso en kilogramos de la paciente.
- **Variación de carga**: Incremento de la tensión nominal de homeostasis.
- **Dosis de denosumab**: Miligramos de fármaco en cada inyección.
- **Lapso entre inyecciones**: Días entre dos inyecciones.

En la tabla 7.1 se muestran los valores máximos y mínimos entre los que se encuentran las variables de entrada. Las diferentes combinaciones se hacen de manera aleatoria entre dichos límites:

Tabla 7.1 Variables de entrada para entrenar la red y sus límites.

Parámetro	Mínimo	Máximo
Edad (años)	58	75
Edad de la menopausia (años)	45	55
K5	0	0.17
Peso (kg)	50	90
Incremento de carga (%)	-10	10
Dosis Dmab (mg)	30	90
Lapso entre inyecciones (días)	90	270

Se crean matrices con números de filas igual a los 250 casos diferentes de combinaciones aleatorias de los 7 parámetros de la tabla 7.1. Se han obtenido un total de 20 matrices diferentes con 250 casos de estudio cada una que se simulan en el modelo matemático.

Del modelo se obtienen las salidas que se pueden ver en la tabla 7.2. Los valores de estas suelen encontrarse entre los rangos mostrados

Tabla 7.2 Variables de salida para entrenar la red y sus rangos.

Parámetro	Descripción	Mínimo	Máximo
Fracción volumétrica de hueso (%)		17	30
Fracción de ceniza		0.65	0.72
Daño		0.0001	1
Oca		1.00×10^{-5}	1.50×10^{-4}

- **Fracción volumétrica de hueso**: Es la medida que se utiliza para evaluar la cantidad de hueso que hay en relación con el volumen total de tejido.

- **Fracción de ceniza:** Mide el contenido mineral del hueso como se vio en el capítulo 2.
- **Daño:** Es una medida de la pérdida de rigidez en el hueso y se encuentra en el intervalo [0,1], donde 0 se corresponde con un hueso sin daño y 1 se corresponde con un estado de fractura o fallo del hueso.
- **Número de OCa:** Concentración de osteoclastos activos.

Para obtener los diferentes valores de salida, se ejecuta el modelo para las combinaciones aleatorias de parámetros de entrada. El modelo simula tres intervalos de tiempo distintos; el envejecimiento, representado por la influencia de la edad, la menopausia y el tratamiento, que se define mediante los dos parámetros de entrada: dosis de denosumab y frecuencia del tratamiento. Se expresa de manera visual el proceso en la figura 7.2.

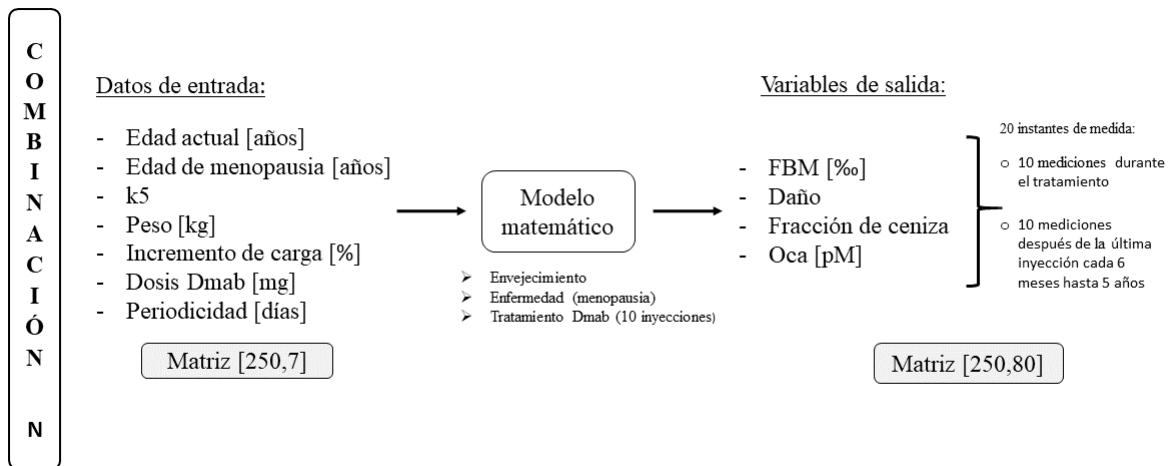


Figura 7.2 Proceso de recogida de datos de salida del modelo matemático.

En total se realizan veinte mediciones de los valores de las variables de salida. Parte de estas se toman en la etapa de tratamiento y la otra en la etapa de después de este. En la etapa de tratamiento se asume que se administra un número constante de 10 inyecciones de denosumab. Las primeras diez mediciones corresponden al día previo a cada inyección. La variable "lapso entre inyecciones" indica el número de días entre dos inyecciones y por ende la duración total del tratamiento.

Una vez finalizadas las diez inyecciones, las siguientes diez mediciones se toman cada seis meses durante los próximos cinco años. De esta manera se obtienen los veinte puntos de medida de los parámetros de salida de la tabla 7.2.

Por lo tanto, al final de cada combinación de parámetros de entrada se obtienen ochenta mediciones distintas, veinte por cada variables (fracción volumétrica de hueso, fracción de ceniza, daño y OCa). Para agilizar el proceso de recogida de estos datos se ha creado una función de Matlab. Para utilizar esta función es necesario preparar el main y la rutina para la introducción de las diferentes variables de entrada. Dicha preparación es similar a las ya comentadas durante este trabajo en el punto 5.4.2, donde se igualan las variables necesarias a las específicas de cada caso. Una vez realizado dicho ajuste, la función contiene un bucle para los 250 casos de cada matriz correspondiente. Los datos de salida se acumulan en una matriz de [250,80].

Una vez realizadas las veinte simulaciones para la obtención de las 20 matrices que contienen las mediciones, ya se poseen todos los datos de salida necesarios para entrenar las diferentes redes neuronales.

Para entrenar las redes es necesario unificar las 20 matrices en una sola matriz tanto para el caso de la matriz de entrada como el de la matriz de salida. Además, en el caso de la matriz de salida, se

separan las 20 mediciones de cada parámetro obteniendo así cuatro matrices de salida diferentes. Siendo las dimensiones de las matrices: matriz de entrada a la red: [5000,7] y matriz de salida de la red: [5000,20] respectivamente. En todas las matrices cada fila representa un caso de estudio diferente, mientras que cada columna en la matriz de entrada representa las variables específicas de cada paciente y en el caso de la matriz de salida representa cada una de las 20 mediciones. Se obtienen así los conjuntos de datos necesarios para el entrenamiento de las cuatro redes neuronales (ver figura 7.3 y 7.4). Obteniendo una matriz de entrada que es la misma para todas las redes y cuatro matrices diferentes de salida para fracción volumétrica de hueso, fracción de ceniza, daño y OCa con las 20 mediciones correspondientes.

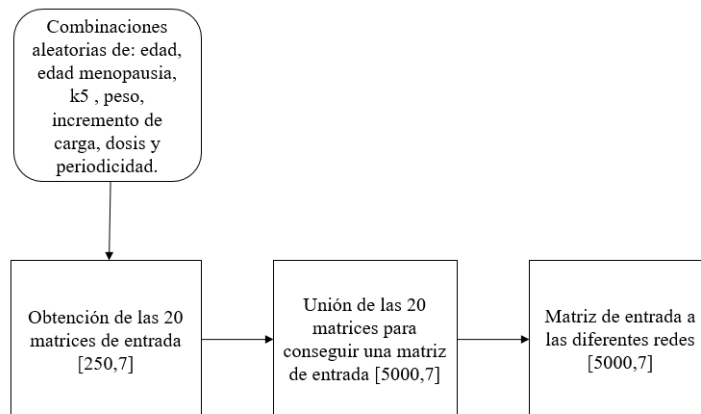


Figura 7.3 Proceso para la obtención de la matriz de entrada.

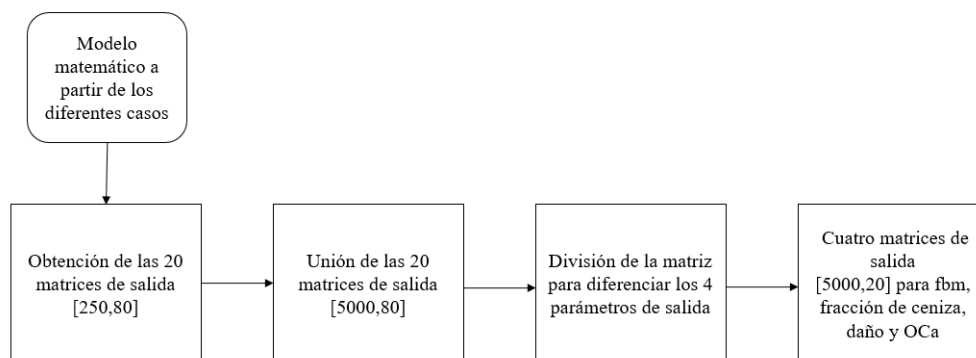


Figura 7.4 Proceso para obtención de las matrices de salida.

Diseño de las redes neuronales

Una vez que se han obtenido las matrices de entrada y salida necesarias para el entrenamiento de las redes, se posee toda la información necesaria para crear las mismas. El procedimiento a seguir es el descrito en el punto 6.6. A continuación, se presenta un esquema general que resume lo que se aborda en este punto (ver figura 7.5).

Como se ha comentado anteriormente, con el objetivo de conseguir un tratamiento óptimo personalizado, se han implementado las redes neuronales. Estas aceleran el proceso de obtención de la solución del modelo matemático. Sin ellas, sería necesario acudir al modelo para cada caso

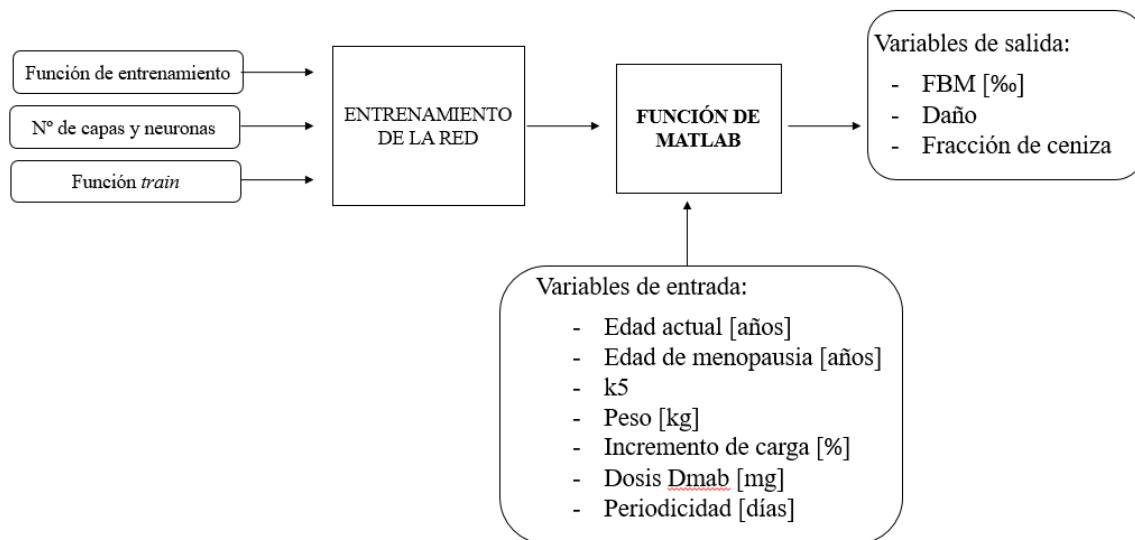


Figura 7.5 Proceso de obtención de variables a predecir por las redes neuronales.

especifico y esto consumiría una gran cantidad de tiempo computacional.

Por lo tanto, se crean tres redes neuronales para predecir los valores de tres parámetros de estudio: fracción volumétrica de hueso, daño y fracción de ceniza. A pesar de que se lleva a cabo la obtención de conjuntos de datos y entrenamiento de la red neuronal relacionada con el parámetro osteoclastos activos, finalmente no se incluye en la optimización de este proyecto. Sin embargo, se ha desarrollado y queda disponible para posibles trabajos futuros. Las tres redes finalmente seleccionadas se utilizan posteriormente en el algoritmo de optimización. El proceso de entrenamiento consiste en:

- Primero, se introducen las matrices de entrada y salida a la red para su entrenamiento. De esos datos, el 75 % se toman para el entrenamiento, 15 % para validación y el 15 % restante para testeo.
- En segundo lugar, se eligen los parámetros de las diferentes redes: número de capas, número de neuronas por capas y función de entrenamiento.
- Tras la elección de los parámetros, se realiza el entrenamiento de las redes a través del Tool box de Matlab.
- Finalmente, se obtiene una función de Matlab (la ilustrada en la figura 7.5 a través de la cual se pueden predecir los valores de los parámetros gracias a la red neuronal creada.

Este proceso de creación de redes se ha realizado en varias fases, probando distintos parámetros de las redes y funciones de entrenamiento. Se puede ver un resumen de los diferentes entrenamientos realizados en la tabla 7.3.

La elección de las redes neuronales de cada parámetro se ha fundamentado principalmente en el análisis de las distintas rectas de regresión. Se han seleccionado las rectas cuyo coeficiente de correlación (R) se acerca más a 1. Esto se debe a que un valor de R cercano a 1 indica una relación altamente lineal y positiva entre las variables en la regresión. En otras palabras, estas rectas son las que mejor se ajustan a los datos objetivos y representan una relación sólida y directamente proporcional entre las variables. En las siguientes figuras 7.6 y 7.7 se pueden apreciar las diferencias entre algunas de las rectas de regresión relacionadas con las variables fracción de ceniza y fracción volumétrica de hueso:

Tabla 7.3 Parámetros de los diferentes entrenamientos.

Entrenamiento 1		
Variables dinámicas	Neuronas y capas	Función de entrenamiento
Fbm	[20,10]	trainlm
Alpha	[20,10]	trainlm
Daño	[20,10]	trainlm
Entrenamiento 2		
Fbm	[40,10]	trainbr
Alpha	[40,10]	trainbr
Daño	[40,10]	trainscg
Entrenamiento 3		
Daño	[40,10]	trainbr

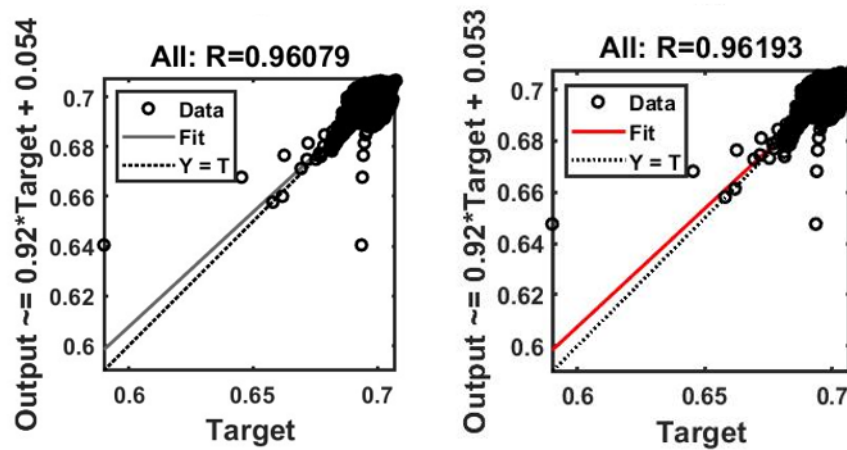


Figura 7.6 Rectas de regresión de la variable fracción de ceniza de los entrenamientos 1 y 2 respectivamente.

Finalmente, se eligen las redes neuronales del entrenamiento 1 para las variables fracción volumétrica de hueso y daño, y la del entrenamiento 2 para la variable fracción de ceniza.

RN_entrenamientos.

Una vez que se han seleccionado las redes neuronales a utilizar, es posible adquirir los valores del modelo matemático necesarios para llevar a cabo el cálculo del tratamiento óptimo.

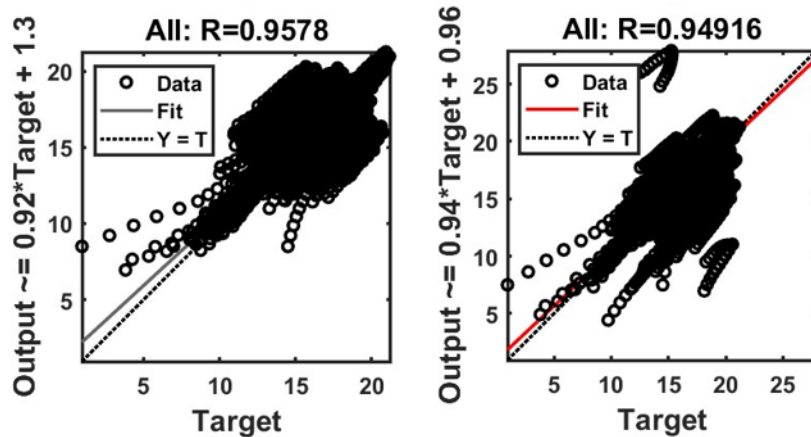


Figura 7.7 Rectas de regresión de la variable fracción volumétrica de huesos entrenamientos 1 y 2 respectivamente.

7.3 Tratamiento óptimo personalizado

El objetivo es determinar un tratamiento óptimo para cada paciente, para ello se utiliza un algoritmo de optimización.

En este punto, se lleva a cabo un enfoque dual de desarrollo. En primer lugar, se procede a explicar detalladamente el proceso de obtención del tratamiento óptimo a través de la utilización de Matlab. En segundo lugar, se aborda la creación y configuración de la interfaz optimizadora, esbozando sus componentes clave y su funcionamiento integral.

7.3.1 Optimización: Implementación en Matlab

Como se ha expuesto previamente en el punto 7.1, se ha optado por emplear la función *fmincon* de Matlab con la finalidad de encontrar el mínimo de la función objetivo escogida. Con el propósito de obtener el tratamiento óptimo deseado, se han desarrollado dos scripts en Matlab destinados a la implementación de la mencionada función *fmincon*.

Definición de la función objetivo y sus restricciones

El primer paso consiste en la definición de la función objetivo, así como de las restricciones asociadas a dicha función.

$$F \text{ optimización} = - \sum_{i=1}^n \frac{FBM_i}{20} + \text{penalizaciones} \quad (7.1)$$

La función objetivo tiene como objetivo principal maximizar la fracción volumétrica de hueso, la cual se define como la media de la fracción volumétrica en todos los instantes de tiempo medidos. Además, se incorporan dos penalizaciones asociadas a los valores de las variables fracción de ceniza (alpha) y daño. Estas penalizaciones se aplican para todos los valores de alpha y daño con el objetivo de que dichos valores se encuentren siempre dentro de la región factible, aumentando la penalización en función de la magnitud de la cercanía a los límites.

$$P_1 = P_{1_0} + \frac{k_1}{(s_{\text{alpha}} - c_{\text{lim}})^{p_{\text{alpha}}}} \quad (7.2)$$

$$P_2 = P_{2_0} + \frac{k_2}{(s_{\text{daño}} - c_{\text{lim}})^{p_{\text{daño}}}} \quad (7.3)$$

Las ecuaciones P_1 y P_2 corresponden a las ecuaciones que modelan las restricciones de fracción de ceniza (α) y daño respectivamente. Las variables P_{1_0} y P_{2_0} representan a los valores iniciales de la penalización, ya que se van acumulando. Las constantes k_1 y k_2 son los factores de penalización para α y daño respectivamente. Las variables s_{α} y $s_{\text{daño}}$ son los valores de α y daño que van a ser evaluados para aplicar la penalización correspondiente. Finalmente, c_{lim} es el límite que se impone a las variables para aplicar la penalización, que es variable en cada caso según la tabla 7.5. Para implementar en Matlab el método de optimización descrito en el punto 7.1.2 se necesita definir un punto inicial. Dicho punto inicial debe estar dentro de la región factible, es decir, los valores de daño y fracción de ceniza deben cumplir con las restricciones. Para definir este punto inicial de partida, se estudian casos aleatorios de pacientes para valores de dosis nula y lapso entre inyecciones máximo. El objetivo de estos valores concretos de dosis y lapso entre inyecciones es analizar los valores de daño y fracción de ceniza en las 20 mediciones para ver para este punto inicial [0,270] qué límites de daño y fracción de ceniza se cumplen.

Se plantean las siguientes combinaciones de límites para alpha y daño:

Tabla 7.4 Límites de Alpha y Daño.

Límite alpha	Límite daño
0.73	0.01
0.73	0.001
0.72	0.01
0.72	0.001
0.71	0.01
0.71	0.001

Una vez analizados 10 casos aleatorios de pacientes se escogen las dos siguientes combinaciones de límites:

Tabla 7.5 Combinaciones finales de estudio.

Alpha < 0.72, Daño < 0.01
Alpha < 0.71, Daño < 0.01

Estas dos combinaciones de límites se escogen en función de los valores de alpha y daño en las 20 mediciones para los casos de estudio. Se muestra un caso de ejemplo con los datos de partida de la tabla 7.6 y los resultados de la tabla 7.7.

Tabla 7.6 Datos del Paciente.

Edad actual	70
Edad de la menopausia	55
Peso (kg)	65
T-score	3.5
Incremento de carga (%)	2
Dosis (mg)	0
Lapso entre inyecciones (días)	270

Como puede verse en la tabla 7.7 los valores del daño para las mediciones 12 hasta la 20 son mayores de 0.001. Además de ocurrir en este caso, que es uno de los 10 realizados, ocurre en otros 3 más. Por ello se fija el límite de daño en 0.01. Sin embargo, en el caso de alpha en la primera columna de

Tabla 7.7 Medición de Alpha y Daño.

Medición	Alpha	Daño
Durante el tratamiento		
1	0.695423056	0.000643134
2	0.69564960	0.000807118
3	0.695757268	0.000625962
4	0.695793183	0.00065282
5	0.695776832	0.000529561
6	0.695751673	0.000724211
7	0.695711788	0.000527678
8	0.69566030	0.000646354
9	0.695614658	0.000598352
10	0.695525178	0.000627592
Después del tratamiento		
11	0.695561889	0.000851037
12	0.694830945	0.001027175
13	0.694367441	0.00123015
14	0.693948471	0.001836313
15	0.693574681	0.002151224
16	0.69322580	0.002470226
17	0.692882534	0.00301485
18	0.692515558	0.00355190
19	0.692166912	0.004186578
20	0.692026657	0.008037493

la tabla 7.7 puede verse como se cumplen sin problemas los límites de 0.73, 0.72 y 0.71 planteados en la tabla 7.4. Por ello se escogen los dos más restrictivos. Quedando así dos combinaciones con el mismo límite de daño y dos límites de alpha distintos tal como se ve en la tabla 7.5.

Tras el análisis explicado se determina que el punto inicial de partida para la optimización es un valor nulo para la dosis y el valor máximo de lapso entre inyecciones de 270 días. Con este punto inicial, los valores de alpha y daño se encuentran dentro de la región factible.

Para conservar los valores de la fracción de ceniza y el daño dentro de la región factible se definen las funciones de penalización 7.2 y 7.3. Estas funciones de penalización deben ser positivas en la región factible y tender a infinito cuando se acerquen a la frontera. Para que así cuando se aleje de la frontera no sea muy penalizado y cuando se acerque a ella sí. De esta manera la búsqueda de la solución óptima siempre se realiza dentro de la región factible.

Se definen las funciones de penalización con los siguientes valores de los parámetros que la describen:

Tabla 7.8 Parámetros de las funciones de penalización para Alpha y Daño.

		k	p
Función de penalización para Alpha	Alpha < 0.72 y Daño < 0.01	0.0001	2
	Alpha < 0.71 y Daño < 0.01	0.00001	2
Función de penalización para Daño	Alpha < 0.72 y Daño < 0.01	0.001	2
	Alpha < 0.71 y Daño < 0.01	0.0001	2

Se representan a continuación en las figuras 7.8 y 7.9 las funciones de penalización de alpha y daño respectivamente para ambas combinaciones de límites:

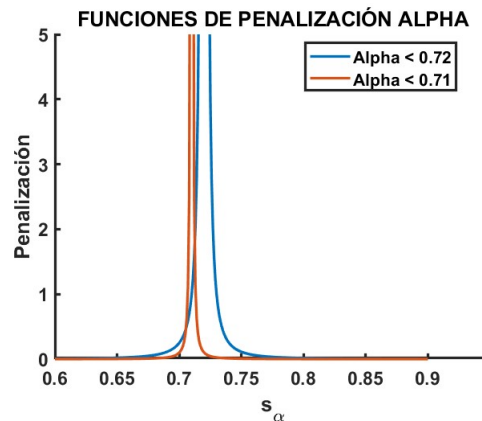


Figura 7.8 Funciones de penalización para la variable alpha.

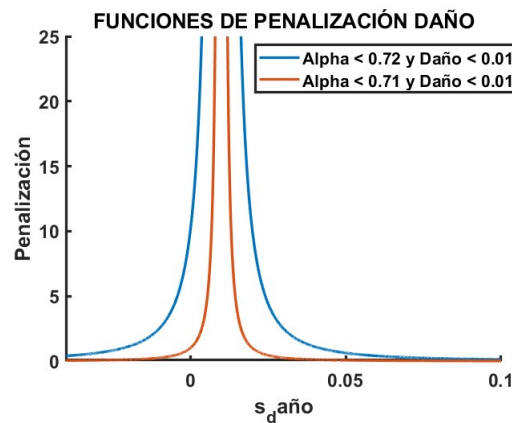


Figura 7.9 Funciones de penalización para la variable daño.

En ambas figuras 7.8 y 7.9 se puede ver como la penalización aumenta a mayor cercanía del valor límites, y como dentro de la región factible la penalización disminuye.

Mediante el proceso descrito, se ha logrado definir la función objetivo a optimizar, así como las funciones de penalización pertinentes para el método del punto interior, haciendo uso de la función de Matlab *fmincon*. Además, se ha establecido una solución inicial específica con el propósito de asegurar que se encuentre dentro de la región factible. Este enfoque se ha aplicado a dos casos de estudio particulares, los cuales han sido delimitados en función de los límites para las variables Alpha y Daño.

7.3.2 Definición de los parámetros de *fmincom*

Una vez se definen la función objetivo, sus restricciones, el punto inicial y los casos de estudio, ya se puede desarrollar la implementación en Matlab del proceso de optimización.

Para llevar a cabo este proceso en Matlab, se han implementado dos funciones: *fecuaciones_OPTIMIZADOR* y *OP_I_optimizador*. La primera de ellas es el script principal, en el cual se realiza la definición y llamada a la función *fmincon*. La segunda función, por su parte, se encarga de definir la función objetivo gestionar las llamadas a las redes neuronales y realizar las penalizaciones correspondientes. Estas función se detallan en profundidad en este punto.

El objetivo es que la función principal reciba los parámetros específicos que definen a cada paciente

introducidos en la interfaz, y gracias a estos parámetros se realice la optimización correspondiente. Como resultado de este proceso, se obtienen los valores óptimos de dosis y lapso para cada paciente. En primer lugar se definen las parámetros característicos de la función *fmincon*:

```

1
2 options = optimoptions('fmincon','PlotFcn',[],...
3                       'Display','iter','OptimalityTolerance',1e-
4                       5,'MaxIterations',1000,'OutputFcn');
5
6 ecuacion= @(P)feccionesOPTIMIZADOR(P,C,o);
7
8 [z,fvalga,exitflag] = fmincon(ecuacion,init_sol,Ares,...
9                             bres,[],[],limite_inferior,limite_superior,[],options);

```

1. Se definen las opciones de optimización utilizando 'optimoptions'. En este caso, se han configurando varias opciones:
 - OptimalityTolerance,1e-5: Establece la tolerancia para la optimización.
 - MaxIterations,1000: Limita el número máximo de iteraciones a 1000.
2. Se define una función "ecuacion" (ver código de arriba) que toma un vector "P" como entrada y devuelve el valor de la función objetivo a ser minimizada.
3. Se utiliza *fmincon* para llevar a cabo la optimización. Los argumentos de esta función son los siguientes:
 - "ecuacion": La función objetivo a minimizar (Eq. 7.1).
 - init_sol: La solución inicial, es decir, el punto de partida para la búsqueda de la solución óptima. El punto concreto es [Dosis,Lapso] = [0,270].
 - "Ares", "bres": Matrices y vectores que definen las restricciones lineales en forma estándar, que este caso no hay.
 - "limite_inferior" y "limite_superior": Vectores que definen los límites inferiores y superiores de las variables de decisión. Siendo, [dosis,lapso], límite_inferior = [30,90] y límite_superior = [90,270]
 - "options": Las opciones de optimización configuradas previamente.
4. La función *fmincon* devuelve varios resultados:
 - "z": La solución óptima encontrada.
 - "fvalga": El valor de la función objetivo en la solución óptima.
 - "exitflag": Un indicador que informa sobre el motivo por el cual la optimización pudo terminar..

La denominada "ecuacion", según se ha definido en la configuración de la función *fmincon*, está precisamente definida en el *fecciones_OPTIMIZADOR* y que posteriormente se evalúa en *OP_1_optimizador*.

A continuación, se describen las dos funciones de Matlab que realizan el proceso de optimización:
Desarrollo script fecciones_OPTIMIZADOR

El objetivo de de esta función es definir la función objetivo de tal manera que se pueda evaluar según los datos de la paciente concreta insertados en el formulario.

Para ello se definen dos vectores:

- **P**: Vector de incógnitas [dosis,lapso]. Para el punto inicial es dosis nula y lapso 270 días.
- **C**: Vector de constantes, que son las variables recogidas de la interfaz, personalizadas en el caso de cada paciente.

A continuación, se llama a las funciones de las Redes Neuronales entrenadas, para que estas proporcionen la predicción del resultado de las variables dinámicas correspondientes (fbm, fracción de ceniza y daño) para el caso concreto de estudio. Las variables dinámicas se almacenan en los denominados vectores "s" de salida. En estos vectores están los valores para las 20 mediciones en el tiempo. Una vez se han predicho las variables dinámicas, se procede a diseñar las penalizaciones. Se inicializan en cero y se definen los factores de penalización. Para acabar se recorren los valores de fracción de ceniza y daño obtenidos de las redes neuronales para evaluar su penalización. El sumatorio de dichas penalizaciones aparece en la función objetivo P_{obj} . Finalmente, se define la función objetivo siendo la fracción volumétrica a maximizar la media entre las 20 mediciones obtenidas de la red neuronal.

Con esta función se evalúa la función objetivo en cada iteración en función de los parámetros de la interfaz y gracias a las predicciones de las redes.

Desarrollo script OP_1_optimizador

Una vez que se ha proporcionado la definición de la función *fmincon* y la función de Matlab *fecuaciones_OPTIMIZADOR*, se procede a explicar el script en el cual se lleva a cabo su implementación, denominado *OP_1_optimizador*. Dicho script es una función. Se establece esta función con el propósito de seguir la dinámica de la interfaz previamente descrita en el capítulo 4. En este contexto, la función recibe los parámetros necesarios de la paciente, los cuales han sido ingresados mediante la interfaz optimizadora. Estos aspectos se abordan con mayor profundidad en el punto siguiente 7.3.3.

Entradas a la función:

- Edad Actual de la paciente
- Edad de la menopausia
- Parámetro T-score
- Peso de la paciente
- Incremento de carga
- Límite de alpha
- Límite de daño

Salidas de la función:

- Dosis de denosumab
- Lapso entre inyecciones
- Fracción volumétrica óptima

Se procede a definir los límites superiores e inferiores de las variables sujetas a optimización, las cuales corresponden a la dosis y al lapso entre inyecciones. En este contexto, es importante señalar que no se imponen restricciones entre las incógnitas, lo cual se refleja en la asignación de valores nulos a los parámetros "Ares" y "bres". A continuación, se organizan y estructuran en el vector "C" los valores previamente recopilados de la interfaz, que son las entradas de la función.

Se lleva a cabo el proceso de optimización llamada a *fmincon*. Se define esta, tal como se ha explicado anteriormente, y se establece una solución inicial para las variables de dosis y lapso. Al concluir,

se obtienen los valores de dosis y periodicidad. Se vuelve a llamar a la función *fecuaciones_OPTIMIZADOR* con los valores óptimos para conocer los valores de penalización de alpha y daño.

Finalmente, se calcula la fracción volumétrica óptima restándole las penalizaciones aplicadas.

7.3.3 Optimización: Diseño y desarrollo de la interfaz en App Designer

Siguiendo el mismo procedimiento que en el capítulo 5, se desarrolla ahora la interfaz optimizadora, ver figura 7.10. En el Design View se añaden los componentes para recopilar los datos de la paciente:

- **Campos de texto:** Se utilizan para ingresar información como código de la paciente, etc. Los campos de texto permiten al usuario escribir texto.
- **Campos numéricos:** Se utilizan para ingresar información de la paciente: la edad actual, la edad de la menopausia, el peso, el T-score y el incremento de carga.
- **Botón:** Se agrega un botón de "Optimizar" en la interfaz de App Designer para indicar la finalización del proceso de ingreso de datos y activar el código interno de a interfaz para obtener el tratamiento óptimo.
- **Campo de lectura:** Se definen unos campos inicialmente vacíos donde una vez obtenido el tratamiento se muestran los valores de este en ellos.
- **Desplegables:** Se utilizan para seleccionar el valor del límite de los parámetros sujetos a restricciones. En este caso concreto las dos combinaciones de estudio son las de la tabla 7.5.
- **Paneles:** Se utilizan para organizar y estructurar los datos agrupando componentes relacionados y facilitando así la presentación de la interfaz. Se utilizarán los siguientes paneles:
 - Panel de información de la paciente.
 - Panel de información de la optimización.
 - Panel de tratamiento óptimo.

El siguiente paso es asignar nombres descriptivos a cada uno de los elementos en el navegador de componentes, ver figura 7.11.

A continuación, tras configurar los nombres y las propiedades de cada componente en la interfaz de App Designer, el siguiente paso es determinar los callbacks correspondientes a cada componente. En el caso de este código, se utiliza el callback de botón presionado. Este callback se activa cuando se presiona un botón en la interfaz. En el caso específico de la interfaz de estudio, cuando el usuario presiona el botón de "Optimizar", se considera que se ha completado el proceso de relleno de datos del paciente y se inicia la ejecución del código. Las acciones que se llevan a cabo tras pulsar el botón:

- **Fin del relleno de datos:** Al pulsar el botón los datos introducidos son enviados a la función *OP_1_optimizador* para realizar el proceso de optimización.
- **Inicio de la optimización:** Una vez que los datos se consideren correctos, se inicia el proceso de optimización, utilizando los valores ingresados como entradas.
- **Vaciar los campos de entrada:** Una vez terminada la optimización, se borran los campos de ingreso de texto y números, de manera que estén listos para ser rellenos nuevamente con otros datos en futuras interacciones.

Una vez se ha definido que el callback 'botonButtonPushed' se define el código necesario para poder realizar la optimización dentro de este, en la pestaña de Code View (ver figura 7.12). Dentro de este se realiza lo siguiente:

OPTIMIZER

PATIENT'S INFORMATION

Patient code number Weight (kg)

Age T-score

Age of menopause

Increment of load w.r.t. nominal value (5)

LIMIT INFORMATION

Ash fraction limit Damage limit

OPTIMAL TREATMENT

Denosumab dose (mg)

Time between 2 injection (days)

Mean fbm* (%)

* The mean fbm is the mean between the 20 samples during and after the treatment.

Figura 7.10 Vista de diseño de la interfaz optimizadora.

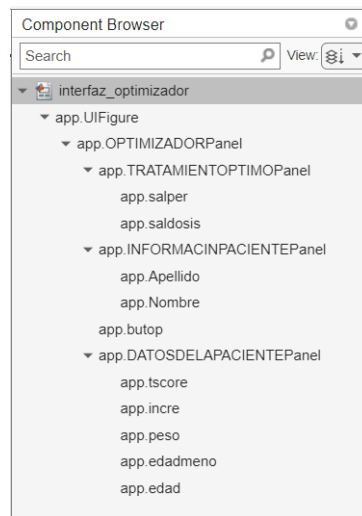


Figura 7.11 Vista del Component Browser con los nombres de los componentes de la interfaz.

- Asignación de variables introducidas en la interfaz y envío de estas al workspace: Al igual que en la interfaz anterior, se recoge la información introducida por el usuario utilizando `app.nombre_variable.Value`. Luego, se puede asignar este valor a una variable llamada "Name" utilizando `Name = string(app.name.Value)`. Una vez se recojan todas las variables, se envían al workspace para poder utilizarlas en las funciones explicadas anteriormente: *fecuaciones_OPTIMIZADOR* y *OP_1_optimizador*. Además, se recogen los valores de los límites de alfa y daño de los despleables de la interfaz.
- Función que optimiza el tratamiento: Una vez se tienen las variables en el workspace ya se puede trabajar con ellas. En el código se llama a la función `valork5` para el cálculo del

parámetro k5 en función del T-score introducido. Tras esto ya se tienen todos los parámetros necesarios para optimizar la función objetivo, por lo que se llama a la función OP_1_-optimizador.

- Relleno de campos de visualización y vaciado de campos de relleno: Una vez se ha realizado la optimización, los valores del caso del tratamiento óptimo, se colocan en los campos de lectura para que el usuario pueda visualizar el tratamiento óptimo de una manera clara en la interfaz. Finalmente se vacían los campos de escritura para que el usuario pueda introducir un nuevo caso.

```

34 % Button pushed function: butop
35 function butopButtonPushed(app, event)
36     %% DATOS INICIALES PARA RELLENAR EL VECTOR C DE PARÁMETROS DE LA PACIENTE
37
38     Edad = app.edad.Value;
39     Edad_meno=app.edadmeno.Value;
40     Peso=app.peso.Value;
41     Tscore=app.tscore.Value;
42     Incre=app.incre.Value;
43
44     assignin('base','Edad_w',Edad); % enviado al workspace
45     assignin('base','Edad_meno_w',Edad_meno); % enviado al workspace
46     assignin('base','Peso_w',Peso); % enviado al workspace
47     assignin('base','Tscore_w',Tscore); % enviado al workspace
48     assignin('base','Incre_w',Incre); % enviado al workspace
49
50     % Llamada a la función que optimiza el tratamiento
51
52     k5 = valorK5(Tscore,Edad,Edad_meno); % Primero se calcula el k5 entrando en la matriz3d con el T-score introducido
53
54     [dosis_op,per_op] = OP_1_optimizador(Edad,Edad_meno,k5,Peso,Incre);
55
56     % Se rellenan los campos de dosis y lapso de tiempo entre inyecciones con el valor optimizado
57
58     app.saldosis.Value = num2str(dosis_op);
59     app.salper.Value = num2str(per_op);
60
61     % Se ponen las casillas de los datos a cero
62
63     app.edad.Value = 0;
64     app.edadmeno.Value = 0;
65     app.peso.Value = 0;
66     app.tscore.Value = 0;
67     app.incre.Value = 0;
68

```

Figura 7.12 Code View de la interfaz optimizadora..

7.4 Resultados

Después de comprender el funcionamiento interno de la interfaz y el proceso de optimización realizado, se procede a llevar a cabo dos casos de estudio específicos con el fin de analizar sus resultados.

Para lograrlo, se define un caso nominal y se varía el valor de cada parámetro en dos valores superior e inferior:

Tabla 7.9 Caso nominal con sus respectivas variaciones.

Parámetro	Caso nominal	Variación 1	Variación 2
Edad Actual	61 años	65 años	57 años
Edad de la Menopausia	50 años	55 años	45 años
Peso	60 kg	70 kg	50 kg
T-score	2.6	3	2
Variación de Carga	0	5	-5

Por lo tanto, en base al caso nominal, se procede a variar un parámetro, como por ejemplo la edad, en dos valores diferentes, uno por encima y otro por debajo de su valor original, manteniendo constante el resto de los parámetros del caso nominal. Esto permite analizar de manera individual la influencia de cada parámetro en el tratamiento. Con el propósito de esclarecer esta metodología, se presentan tablas que incluyen los valores del parámetro que se modifica, así como los resultados de dosis, lapso entre inyecciones y fracción volumétrica óptima de hueso para dicho tratamiento específico.

Se exponen y comentan a continuación los resultados para $\alpha < 0.72$ y daño < 0.01 :

Tabla 7.10 Influencia de la edad para: $\alpha < 0.72$ y Daño < 0.01 .

Edad	Dosis	Lapso	Fracción volumétrica
65	90.00	252.77	17.37
60	90.00	185.24	17.90
57	90.00	142.24	18.39

Influencia de la edad: A mayor T-score más grave es la osteoporosis. Dos mujeres con diferente edad y el mismo valor de T-score implica que la persona más joven tiene una enfermedad más severa. Como la paciente de 57 años tiene una enfermedad más grave necesita las dosis más frecuentes, por eso el lapso es menor para la de 57 años. El caso nominal tiene un lapso intermedio entre las mujeres de 65 y 57 años. A mayor edad y mismo T-score la enfermedad es menos grave y por eso el lapso es más grande. Se puede ver también en la tabla 7.10 como las tres edades tienen la dosis máxima, 90 mg. Como el objetivo del optimizador es maximizar la fracción volumétrica, esto se consigue con el valor de dosis límite.

Tabla 7.11 Influencia de la edad de la menopausia para: $\alpha < 0.72$ y Daño < 0.01 .

Edad de la menopausia	Dosis	Lapso	Fracción volumétrica
55	90.00	172.03	18.09
50	90.00	185.24	17.90
45	51.08	90.00	17.46

Influencia de la edad de la menopausia: Si la menopausia es antes, lo normal es tener un T-score mayor por tener menos densidad ósea. No es el caso de estudio ya que se ha forzado el mismo T-score. Si para diferentes edades de menopausia se tiene el mismo T-score significa que se ha producido una bajada tan significativa de la densidad como se esperaba y por tanto la paciente con menopausia temprana no ha sufrido una bajada de masa ósea tan grave con respecto a la paciente que comenzó la menopausia a los 55 años. Por ello la de 45 años, necesita menos dosis ya que si su enfermedad no ha sido tan grave no es necesario. Además de la reducción en la dosis, se produce también una disminución en el lapso entre las inyecciones. Esta disminución del lapso entre inyecciones puede deberse a que, al reducir la dosis, se disminuye la influencia del tratamiento, lo que a su vez puede requerir una administración más frecuente para mantener su efectividad.

Tabla 7.12 Influencia del peso para: $\alpha < 0.72$ y Daño < 0.01 .

Peso	Dosis	Lapso	Fracción volumétrica
70	90.00	164.28	17.79
60	90.00	185.24	17.90
50	89.41	188.52	17.95

Influencia del peso: A mayor peso mayor dosis, por ello disminuye el lapso entre inyecciones al aumentar el peso.

Influencia del T.score: A mayor T-score más grave es la enfermedad y por lo tanto mayor dosis se necesita por ello a mayor valor de T-score más cerca se suministran las inyecciones, más bajo es el lapso.

Influencia de la variación de carga: Los resultados de la optimización no son los esperados. Lo

Tabla 7.13 Influencia del T.score para: Alpha < 0.72 y Daño < 0.01.

T-score	Dosis	Lapso	Fracción volumétrica
3.0	90.00	90.00	17.38
2.6	90.00	185.24	17.90
2.0	84.48	212.87	18.96

Tabla 7.14 Influencia de la variación de carga para: Alpha < 0.72 y Daño < 0.01.

Variación de carga	Dosis	Lapso	Fracción volumétrica
5	90.00	183.92	18.00
0	90.00	185.24	17.90
-5	90.00	181.44	17.78

normal es que el ejercicio implique una reducción en la dosis requerida. Sin embargo, en los tres casos, se observa que la variación de carga no parece tener un impacto significativo la dosis es la máxima y el lapso apenas varia. Esta discrepancia plantea la necesidad de realizar un análisis más profundo en el futuro.

En el caso de los límites $\alpha < 0.71$ y daño < 0.01, al reducir el límite de la fracción de ceniza, se impone una restricción más el contenido mineral. Esta restricción resulta en una disminución de la dosis o un aumento del lapso entre inyecciones. Esta restricción más rigurosa del contenido mineral tiene como objetivo evitar un incremento excesivo de mineralización. A medida que aumenta la cantidad de denosumab en el cuerpo, se observa una mayor supresión de la remodelación ósea y un incremento en el contenido mineral en el tejido óseo. Para prevenir una mineralización excesiva, se implementa una reducción en la dosis.

Tabla 7.15 Influencia de la edad para: Alpha < 0.71 y Daño < 0.01.

Edad	Dosis	Lapso	Fracción volumétrica
65	30.00	197.23	17.03
60	71.88	218.11	17.28
57	81.68	204.89	17.91

Influencia de la edad: Al igual que para Alpha < 0.72, a mayor edad y mismo T-score la enfermedad es menos grave y por ello se necesita menos dosis. Por eso la dosis va disminuyendo a medida que aumenta la edad de la paciente. A diferencia del caso anterior las dosis no son igual ni la máxima, con el objetivo de restringir más el contenido mineral.

Tabla 7.16 Influencia de la edad de la menopausia para: Alpha < 0.71 y Daño < 0.01.

Edad de la menopausia	Dosis	Lapso	Fracción volumétrica
55	75.20	211.99	17.45
50	71.88	218.11	17.28
45	35.97	90.00	17.14

Influencia de la edad de la menopausia: Al igual que antes, si la menopausia es temprana, lo normal es tener un T-score mayor por tener menos densidad ósea. No es el caso de estudio ya que se ha forzado el mismo T-score. Los pacientes con menopausia temprana no tiene una bajada

de densidad tan grave por lo que la dosis es baja y aumenta a medida que aumenta la edad de la menopausia.

Tabla 7.17 Influencia del peso para: $\text{Alpha} < 0.71$ y Daño < 0.01 .

Peso	Dosis	Lapso	Fracción volumétrica
70	74.95	205.60	17.31
60	71.88	218.11	17.28
50	62.34	231.88	17.14

Influencia del peso: A mayor peso mayor dosis, por ello disminuye el lapso entre inyecciones al aumentar el peso. En este caso a diferencia de límite 0-72, las dosis no son las máximas.

Tabla 7.18 Influencia del T-score para: $\text{Alpha} < 0.71$ y Daño < 0.01 .

T-score	Dosis	Lapso	Fracción volumétrica
3.0	90.00	262.07	16.84
2.6	71.88	218.11	17.28
2.0	30.00	223.83	17.79

Influencia del T.score: Aumenta la dosis a mayor T-score por ser la enfermedad más grave.

Tabla 7.19 Influencia de la variación de carga para: $\text{Alpha} < 0.71$ y Daño < 0.01 .

Variación de carga	Dosis	Lapso	Fracción volumétrica
5	68.44	225.50	17.28
0	71.88	218.11	17.28
-5	71.47	208.47	17.22

Influencia de la variación de carga: Al igual que en el caso anterior, se observa que la variación de carga no ejerce un impacto significativo en el tratamiento. Sin embargo, en contraste con el escenario de $\alpha < 0.72$, en este caso las dosis no alcanzan niveles máximos, y el intervalo entre inyecciones es considerablemente más amplio. Cabe señalar que, a pesar de esta variación en las dosis y los intervalos para los tres valores de carga, las diferencias para los tres valores de variación de carga se consideran insignificantes.

Tras el análisis detallado de la influencia de cada parámetro en ambas combinaciones de límites, se puede concluir lo siguiente: en el caso de un límite de α menos restrictivo, como se muestra en el caso 1, se obtienen combinaciones de dosis y lapso que normalmente representan la dosis máxima para maximizar la fracción volumétrica de hueso. No obstante, cuando se establece un valor más restrictivo para la fracción de ceniza, caso 2, se observan disminuciones en la dosis y/o aumentos en el lapso, con el fin de evitar una mineralización excesiva, aunque esto conlleve una reducción en la fracción volumétrica óptima.

8 Conclusiones y trabajos futuros

Tras haber realizado este trabajo, se han obtenido una serie de conclusiones.

La interfaz desarrollada para predecir el cálculo de fractura es una herramienta simple y visual que facilita el análisis de los resultados sin requerir un conocimiento previo del modelo matemático utilizado. No obstante, esta interfaz aún tiene un potencial futuro significativo, ya que sería beneficioso integrar el valor del FRAX para compararlo con los resultados obtenidos. Además, se podrían agregar nuevas variables a la interfaz en función de parámetros medibles de la paciente. Por lo tanto la interfaz calculadora tienen oportunidades de mejora.

En cuanto a los resultados del optimizador, en general son los esperados. Se pueden discutir estos para la variación de cada uno de los parámetros característicos del paciente:

- **Influencia de la edad:** Para pacientes con diferentes edades y un mismo T-score, la persona más joven generalmente presenta una enfermedad más grave y, por lo tanto, necesita dosis más frecuentes. La dosis disminuye a mayor edad.
- **Edad de la menopausia:** La paciente más joven puede requerir dosis menores, aumentando esta al aumentar la edad de la menopausia. Además se produce una disminución del lapso entre inyecciones no esperada.
- **Peso:** un peso más alto se relaciona con una dosis mayor y, en consecuencia, un intervalo entre inyecciones más corto.
- **T-score:** A mayor T-score, dosis mayor y un lapso entre inyecciones más corto.
- **Variación de carga:** No parece afectar significativamente el tratamiento, ya que las dosis son máximas y el lapso apenas varía en los tres casos, lo que plantea la necesidad de un análisis más profundo en el futuro.

Tras estos resultados, se debe destacar la necesidad de llevar a cabo análisis más exhaustivos en el futuro, especialmente en lo que respecta a la influencia de la edad de la menopausia y la variación de carga. Asimismo, al igual que la interfaz calculadora, el optimizador también posee un potencial de desarrollo y expansión considerable. Es posible incorporar nuevas variables de entrada o mejorar el proceso de optimización mediante la aplicación de distintos métodos de optimización.

Implementación del FRAX en la calculadora del riesgo de fractura.

Estudio del resultado de la variación de carga en el optimizador. Es indiferente la variación en los resultados.

Estudio de los resultados de los valores de lapso entre inyecciones para la influencia de la edad de la menopausia. bajada del lapso entre inyecciones en edad meno

Índice de Figuras

1.1	Coste de la enfermedad.Tomada de [7]	3
1.2	Web del FRAX [9]	4
2.1	Composición de la matriz ósea	8
2.2	Estructura del hueso a nivel macroscópico. Tomada de [12]	8
2.3	Osteoclasto reabsorbiendo superficie ósea. Tomada de [13]	9
2.4	Osteoblastos. Tomada de [13]	9
2.5	Ostetocitos conectados mediante sus prolongaciones. Tomada de [14]	10
2.6	Células de borde	10
2.7	Partes del hueso largo. Tomada de [15]	11
2.8	Ejemplos de los distintos tipos de huesos.	12
2.9	Fases de la actividad de una BMU de un hueso trabecular.Tomada de [20]	14
2.10	Teoría del mecanostato de Frost. Adaptación del hueso a estímulos mecánicos. [88]	14
2.11	Diferencia de porosidad en hueso sano y enfermo. Tomada de [24]	15
2.12	Fracturas típicas a causa de la osteoporosis [26]	15
3.1	Representación esquemática del modelo de población de células óseas: se presentan las etapas de diferenciación de las células óseas junto con las interacciones bioquímicas y biomecánicas.	19
3.2	Representación esquemática del modelo PK para Dmab. El recuadro enmarcado representa el modelo PK de un solo compartimento propuesto por Marathe et al. Se observa el compartimento óseo añadido en [38]. El flujo de Dmab desde el compartimento central hacia el compartimento óseo representa el término de producción en las reacciones de unión competitiva entre RANKL, OPG y Dmab	41
3.3	Representación esquemática del modelo PK propuesto. El recuadro enmarcado representa el modelo PK de un solo compartimento propuesto por Martin et al. [41]. Se puede ver el compartimento óseo añadido. El flujo de Rmab desde el compartimento central al compartimento óseo representa el término de producción en la reacción de unión Scl-SclAb.	42
4.1	Entorno de trabajo: Design view.	44
4.2	Entorno de trabajo: Code view.	44
4.3	Elemento editor de diseño.	45
4.4	Elemento librería de componentes.	45
4.5	Elemento cuadrícula.	45
4.6	Barra de tareas.	45
4.7	Elemento editor de código	46
4.8	Elemento Code Browser	46

4.9	Elemento Layout.	47
4.10	Partes del código del code editor.	48
4.11	Creación de funciones privadas y públicas.	48
4.12	Creación de Callbacks.	49
4.13	Creación de propiedad privadas y públicas	49
4.14	Distintos componentes de la librería.	50
4.15	Componentes necesarios para crear la interfaz de ejemplo que simula calculadora.	51
4.16	Vista del Component Browser con todos los elementos de la interfaz de la calculadora.	51
4.17	Visualización de los cuatro callbacks en el Component Browser y sus partes de códigos correspondientes en el Code View.	52
5.1	Interfaz Calculadora: Elementos de la vista de diseño de la interfaz	58
5.2	Vista del Component Browser con los nombres de los componentes de la interfaz	59
5.3	Vista de diseño	59
5.4	Vista de código	60
5.5	Elementos de la vista de código	60
5.6	Esquema de las Hojas de datos	65
5.7	Curva de homeostasis fbm	66
5.8	Curva de homeostasis densidad	67
5.9	Dimensiones de la matriz 3D (Base de datos)	67
5.10	Proceso de obtención de k_5 a partir de un T-score dado	69
5.11	Entradas y salidas de la función <i>valork5</i>	69
5.12	Proceso para implementar el k_5 en la interfaz gráfica	69
5.13	Parte 1 de la interfaz rellena	71
5.14	Evolución de la densidad y fracción volumétrica	72
5.15	Evolución del daño	72
5.16	Parte 2 de la interfaz rellena	72
5.17	Probabilidad de fractura concreta para la vertebra a tracción y probabilidad de fractura concreta para el fémur a compresión	73
6.1	Neurona artificial. Tomada de [78]	76
6.2	Regla más común de propagación. Tomada de [78]	76
6.3	Diferentes funciones de propagación. Tomada de [78].	77
6.4	Capas de una red neuronal. Tomada de [79].	77
6.5	Monocapa vs multicapa.	78
6.6	Red neuronal recurrente.	78
6.7	Redes Neuronales Convolucionales. Tomada de [81]	79
6.8	Redes Neuronales Recurrentes. Tomada de [81]	79
6.9	Redes Neuronales densas. Tomada de [82]	79
6.10	Red Neuronal feedforward. Tomada de [84]	80
6.11	Red neuronal. Tomada de .	81
7.1	Proceso de obtención de la interfaz	89
7.2	Proceso de recogida de datos de salida del modelo matemático	91
7.3	Proceso para la obtención de la matriz de entrada	92
7.4	Proceso para obtención de las matrices de salida	92
7.5	Proceso de obtención de variables a predecir por las redes neuronales	93
7.6	Rectas de regresión de la variable fracción de ceniza de los entrenamientos 1 y 2 respectivamente	94
7.7	Rectas de regresión de la variable fracción volumétrica de huesos entrenamientos 1 y 2 respectivamente	95
7.8	Funciones de penalización para la variable alpha	98

7.9	Funciones de penalización para la variable daño	98
7.10	Vista de diseño de la interfaz optimizadora	102
7.11	Vista del Component Browser con los nombres de los componentes de la interfaz	102
7.12	Code View de la interfaz optimizadora.	103

Índice de Tablas

3.1	Cantidades dinámicas en las ecuaciones de gobierno	20
3.2	Constantes del modelo, sus valores, unidades y descripción	20
5.1	Valores de los parámetros iniciales: Densidad, Tensión y fbm	66
5.2	Parámetros del ejemplo de la interfaz calculadora	70
5.3	Entradas y salidas de las secciones de la interfaz	71
7.1	Variables de entrada para entrenar la red y sus límites	90
7.2	Variables de salida para entrenar la red y sus rangos	90
7.3	Parámetros de los diferentes entrenamientos	94
7.4	Límites de Alpha y Daño	96
7.5	Combinaciones finales de estudio	96
7.6	Datos del Paciente	96
7.7	Medición de Alpha y Daño	97
7.8	Parámetros de las funciones de penalización para Alpha y Daño	97
7.9	Caso nominal con sus respectivas variaciones	103
7.10	Influencia de la edad para: $\text{Alpha} < 0.72$ y $\text{Daño} < 0.01$	104
7.11	Influencia de la edad de la menopausia para: $\text{Alpha} < 0.72$ y $\text{Daño} < 0.01$	104
7.12	Influencia del peso para: $\text{Alpha} < 0.72$ y $\text{Daño} < 0.01$	104
7.13	Influencia del T.score para: $\text{Alpha} < 0.72$ y $\text{Daño} < 0.01$	105
7.14	Influencia de la variación de carga para: $\text{Alpha} < 0.72$ y $\text{Daño} < 0.01$	105
7.15	Influencia de la edad para: $\text{Alpha} < 0.71$ y $\text{Daño} < 0.01$	105
7.16	Influencia de la edad de la menopausia para: $\text{Alpha} < 0.71$ y $\text{Daño} < 0.01$	105
7.17	Influencia del peso para: $\text{Alpha} < 0.71$ y $\text{Daño} < 0.01$	106
7.18	Influencia del T-score para: $\text{Alpha} < 0.71$ y $\text{Daño} < 0.01$	106
7.19	Influencia de la variación de carga para: $\text{Alpha} < 0.71$ y $\text{Daño} < 0.01$	106

Bibliografía

- [1] E. Seeman and P. D. Delmas, “Bone quality—the material and structural basis of bone strength and fragility,” *New England journal of medicine*, vol. 354, no. 21, pp. 2250–2261, 2006.
- [2] O. Johnell and J. Kanis, “An estimate of the worldwide prevalence and disability associated with osteoporotic fractures,” *Osteoporosis international*, vol. 17, pp. 1726–1733, 2006.
- [3] J. A. Kanis, L. J. Melton, C. Christiansen, C. C. Johnston, N. Khaltsev *et al.*, “The diagnosis of osteoporosis,” *J Bone Miner Res*, vol. 9, no. 8, pp. 1137–1141, 1994.
- [4] J. Kanis, N. Norton, N. Harvey, T. Jacobson, H. Johansson, M. Lorentzon, E. McCloskey, C. Willers, and F. Borgström, “Scope 2021: A new scorecard for osteoporosis in europe,” *Arch Osteoporos*, vol. 16, no. 1, p. 82, Jun 2021.
- [5] I. O. Foundation. (Fecha de acceso: 7 de octubre de 2023) About osteoporosis. [Online]. Available: <https://www.osteoporosis.foundation/patients/about-osteoporosis>
- [6] S. E. de Geriatria y Gerontología. (2022) Osteoporosis en españa - febrero 2022. [Online]. Available: <https://www.segg.es/media/descargas/Osteoporosis-en-Espa%C3%B1a.Feb.2022.pdf>
- [7] GeriatricArea. (2022) La osteoporosis: una enfermedad silenciosa que afecta a casi 3 millones de personas en españa. Consultado el 7 de octubre de 2023. [Online]. Available: <https://www.geriatricarea.com/2022/02/17/la-osteoporosis-una-enfermedad-silenciosa-que-afecta-a-casi-3-millones-de-personas-en-espana/>
- [8] C. G. Vaquero and D. R. Vilaseca, “¿ qué es el frax?: pros y contras,” *Seminarios de la Fundación Española de Reumatología*, vol. 11, no. 3, pp. 100–106, 2010.
- [9] A. (si está disponible). (Año (si está disponible)) Título o descripción del recurso. Consultado el Fecha de consulta. [Online]. Available: <https://frax.shef.ac.uk/FRAX/tool.aspx?lang=sp>
- [10] I. Fernández-Tresguerres Hernández-Gil, M. A. Alobera Gracia, M. d. Canto Pingarrón, and L. Blanco Jerez, “Bases fisiológicas de la regeneración ósea i: Histología y fisiología del tejido óseo,” *Medicina Oral, Patología Oral y Cirugía Bucal (Internet)*, vol. 11, no. 1, pp. 47–51, 2006.
- [11] C. Hellmich, “Microelasticity of bone,” in *Applied micromechanics of porous materials*. Springer, 2005, pp. 289–331.
- [12] W. Kahle, W. Platzer, H. Leonhardt, L. Schnellbacher, and G. Spitzer, *Atlas de anatomía para estudiantes y médicos: Aparato locomotor/por Werner Platzer dibujos de Lothar Schnellbacher*

- y Gerhard Spitzer traducido de la 5a ed. alemana por J. Carreres y C. Iñiguez. Omega, 1998.
- [13] “Osteomielite crónica,” <http://anatpat.unicamp.br/lamosso1.html>, 2015, accedido 14-07-2023.
- [14] “Osteología y metabolismo mineral,” <https://idim.com.ar/blog/wp-content/uploads/2015/04/clases-1-y-2.pdf>, 2015, accedido 14-07-2023.
- [15] “Entorno estudiantil,” https://www.entornoestudiantil.com/los-huesos/google_ignette, 2015, accedido 14-07-2023.
- [16] J. G. Gómez, “El proceso de remodelación ósea,” *Orthotips AMOT*, vol. 4, no. 3, pp. 170–176, 2008.
- [17] H. M. Frost, *Bone remodeling and its relationship to metabolic bone diseases*. Charles C. Thomas Publisher, 1973.
- [18] A. M. Parfitt, “The physiologic and clinical significance of bone histomorphometric data.” *Bone histomorphometry: techniques and interpretation.*, pp. 143–223, 1983.
- [19] R. Martin, “Toward a unifying theory of bone remodeling,” *Bone*, vol. 26, no. 1, pp. 1–6, 2000.
- [20] P. Fridez, “Modélisation de l’adaptation osseuse externe,” *EPFL, Physics Department*, 1996.
- [21] F. HM, “The mechanostat: a proposed pathogenic mechanism of osteoporoses and the bone mass effects of mechanical and non mechanical agents,” *Bone Miner*, vol. 2, pp. 73–85, 1987.
- [22] M. Martin, V. Sansalone, D. M. Cooper, M. R. Forwood, and P. Pivonka, “Mechanobiological osteocyte feedback drives mechanostat regulation of bone in a multiscale computational model,” *Biomechanics and modeling in mechanobiology*, vol. 18, no. 5, pp. 1475–1496, 2019.
- [23] J. González Macías and J. Olmos Martínez, “Fisiopatología de la osteoporosis y mecanismo de acción de la pth,” *Rev Osteoporos Metab Miner*, vol. 2, no. Supl 2, pp. S5–S17, 2010.
- [24] “Osteomielite crónica,” <https://www.podoactiva.com/blog/osteoporosis-asi-afecta-la-biomecanica-de-los-pies>, 2015, accedido 03-10-2023.
- [25] T. D. Rachner, S. Khosla, and L. C. Hofbauer, “Osteoporosis: now and the future,” *The Lancet*, vol. 377, no. 9773, pp. 1276–1287, 2011.
- [26] “Osteomielite crónica,” <https://www.infosalus.com/salud-investigacion/noticia-aumentar-consumo-calcio-llevar-vida-activa-reduciria-fracturas-oseas-osteoporosis-20101118135220.html>, 2015, accedido 03-10-2023.
- [27] M. Hermoso de Mendoza, “Clasificación de la osteoporosis: Factores de riesgo. clínica y diagnóstico diferencial,” in *Anales del sistema sanitario de Navarra*, vol. 26. SciELO Espana, 2003, pp. 29–52.
- [28] C. Tapia, “Papel de los estrógenos en la patogenia de la osteoporosis posmenopáusica.”
- [29] S. R. Cummings, J. S. Martin, M. R. McClung, E. S. Siris, R. Eastell, I. R. Reid, P. Delmas, H. B. Zoog, M. Austin, A. Wang *et al.*, “Denosumab for prevention of fractures in postmenopausal women with osteoporosis,” *New England Journal of Medicine*, vol. 361, no. 8, pp. 756–765, 2009.
- [30] F. Cosman, D. B. Crittenden, J. D. Adachi, N. Binkley, E. Czerwinski, S. Ferrari, L. C. Hofbauer, E. Lau, E. M. Lewiecki, A. Miyauchi *et al.*, “Romosozumab treatment in postmenopausal women with osteoporosis,” *New England Journal of Medicine*, vol. 375, no. 16, pp. 1532–1543, 2016.

- [31] J. Martínez-Reina, J. Ojeda, and J. Mayo, “On the use of bone remodelling models to estimate the density distribution of bones. uniqueness of the solution,” *Plos one*, vol. 11, no. 2, p. e0148603, 2016.
- [32] J. L. Calvo-Gallego, F. Gutiérrez-Millán, J. Ojeda, M. Á. Pérez, and J. Martínez-Reina, “The correlation between bone density and mechanical variables in bone remodelling models: Insights from a case study corresponding to the femur of a healthy adult,” *Mathematics*, vol. 10, no. 18, p. 3367, 2022.
- [33] J. Reina, J. M. García-Aznar, J. Domínguez, and M. Doblaré, “Numerical estimation of bone density and elastic constants distribution in a human mandible,” *Journal of Biomechanics*, vol. 40, no. 4, pp. 828–836, 2007.
- [34] J. Ojeda, J. Martínez-Reina, J. García-Aznar, J. Domínguez, and M. Doblaré, “Numerical simulation of bone remodelling around dental implants,” *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 225, no. 9, pp. 897–906, 2011.
- [35] J. Martínez-Reina, J. García-Aznar, J. Domínguez, and M. Doblaré, “A bone remodelling model including the directional activity of bms,” *Biomechanics and modeling in mechanobiology*, vol. 8, pp. 111–127, 2009.
- [36] J. Martínez-Reina, I. Reina, J. Domínguez, and J. García-Aznar, “A bone remodelling model including the effect of damage on the steering of bms,” *Journal of the mechanical behavior of biomedical materials*, vol. 32, pp. 99–112, 2014.
- [37] J. Martínez-Reina and P. Pivonka, “Effects of long-term treatment of denosumab on bone mineral density: insights from an in-silico model of bone mineralization,” *Bone*, vol. 125, pp. 87–95, 2019.
- [38] J. Martínez-Reina, J. L. Calvo-Gallego, M. Martin, and P. Pivonka, “Assessment of strategies for safe drug discontinuation and transition of denosumab treatment in pmc—insights from a mechanistic pk/pd model of bone turnover,” *Frontiers in Bioengineering and Biotechnology*, vol. 10, p. 886579, 2022.
- [39] J. Martínez-Reina, J. L. Calvo-Gallego, and P. Pivonka, “Combined effects of exercise and denosumab treatment on local failure in post-menopausal osteoporosis—insights from bone remodelling simulations accounting for mineralisation and damage,” *Frontiers in Bioengineering and Biotechnology*, vol. 9, p. 635056, 2021.
- [40] J. L. Calvo-Gallego, P. Pivonka, R. Ruiz-Lozano, and J. Martínez-Reina, “Mechanistic pk-pd model of alendronate treatment of postmenopausal osteoporosis predicts bone site-specific response,” *Frontiers in Bioengineering and Biotechnology*, vol. 10, p. 940620, 2022.
- [41] M. Martin, V. Sansalone, D. M. Cooper, M. R. Forwood, and P. Pivonka, “Assessment of romosozumab efficacy in the treatment of postmenopausal osteoporosis: results from a mechanistic pk-pd mechanostat model of bone remodeling,” *Bone*, vol. 133, p. 115223, 2020.
- [42] P. Pivonka, P. R. Buenzli, S. Scheiner, C. Hellmich, and C. R. Dunstan, “The influence of bone surface availability in bone remodelling—a mathematical model including coupled geometrical and biomechanical regulations of bone cells,” *Engineering Structures*, vol. 47, pp. 134–147, 2013.
- [43] J. Lemaitre and J.-L. Chaboche, *Mechanics of solid materials*. Cambridge university press, 1994.
- [44] J. Martínez-Reina, J. García-Aznar, J. Domínguez, and M. Doblaré, “On the role of bone damage in calcium homeostasis,” *Journal of theoretical biology*, vol. 254, no. 3, pp. 704–712, 2008.

- [45] C. Pattin, W. Caler, and D. Carter, "Cyclic mechanical property degradation during fatigue loading of cortical bone," *Journal of biomechanics*, vol. 29, no. 1, pp. 69–79, 1996.
- [46] J. M. Garcia-Aznar, T. Rüberg, and M. Doblare, "A bone remodelling model coupling microdamage growth and repair by 3d bmu-activity," *Biomechanics and modeling in mechanobiology*, vol. 4, no. 2-3, pp. 147–167, 2005.
- [47] R. C. Juvinall, "Engineering considerations of stress, strain, and strength," (*No Title*), 1967.
- [48] J. D. Currey, "Tensile yield in compact bone is determined by strain, post-yield behaviour by mineral content," *Journal of biomechanics*, vol. 37, no. 4, pp. 549–556, 2004.
- [49] M. C. Peterson and M. M. Riggs, "A physiologically based mathematical model of integrated calcium homeostasis and bone remodeling," *Bone*, vol. 46, no. 1, pp. 49–63, 2010.
- [50] R. Ruiz-Lozano, J. L. Calvo-Gallego, P. Pivonka, M. McDonald, and J. Martínez-Reina, "An in-silico approach to elucidate the pathways leading to primary osteoporosis: age-related vs. postmenopausal *En revisión*," 2023.
- [51] V. Nicolas, A. Prewett, P. Bettica, S. Mohan, R. Finkelmann, D. Baylink, and J. Farley, "Age-related decreases in insulin-like growth factor-i and transforming growth factor-beta in femoral cortical bone from both men and women: implications for bone loss with aging," *The Journal of Clinical Endocrinology & Metabolism*, vol. 78, no. 5, pp. 1011–1016, 1994.
- [52] V. Lemaire, F. L. Tobin, L. D. Greller, C. R. Cho, and L. J. Suva, "Modeling the interactions between osteoblast and osteoclast activities in bone remodeling," *Journal of theoretical biology*, vol. 229, no. 3, pp. 293–309, 2004.
- [53] B. L. Riggs, L. J. Melton III, R. A. Robb, J. J. Camp, E. J. Atkinson, J. M. Peterson, P. A. Rouleau, C. H. McCollough, M. L. Bouxsein, and S. Khosla, "Population-based study of age and sex differences in bone volumetric density, size, geometry, and structure at different skeletal sites," *Journal of Bone and Mineral Research*, vol. 19, no. 12, pp. 1945–1954, 2004.
- [54] Y. Gotoh, Y. Okamoto, O. Uemura, N. Mori, S. Tanaka, T. Ando, and M. Nishida, "Determination of age-related changes in human soluble interleukin 2 receptor in body fluids of normal subjects as a control value against disease states," *Clinica chimica acta*, vol. 289, no. 1-2, pp. 89–97, 1999.
- [55] J. Pfeilschifter, I. Diel, B. Scheppach, A. Bretz, R. Krempien, J. Erdmann, G. Schmid, N. Reske, H. Bismar, T. Seck *et al.*, "Concentration of transforming growth factor beta in human bone tissue: relationship to age, menopause, bone turnover, and bone volume," *Journal of Bone and Mineral Research*, vol. 13, no. 4, pp. 716–730, 1998.
- [56] J. L. Calvo-Gallego, P. Manchado-Morales, P. Pivonka, and J. Martínez-Reina, "Spatio-temporal simulations of bone remodelling using a bone cell population model based on cell availability," *Frontiers in Bioengineering and Biotechnology*, vol. 11, p. 1060158, 2023.
- [57] M.-S. M. Ardawi, H. A. Al-Kadi, A. A. Rouzi, and M. H. Qari, "Determinants of serum sclerostin in healthy pre-and postmenopausal women," *Journal of Bone and Mineral Research*, vol. 26, no. 12, pp. 2812–2822, 2011.
- [58] U. I. Mödder, K. A. Hoey, S. Amin, L. K. McCready, S. J. Achenbach, B. L. Riggs, L. J. Melton III, and S. Khosla, "Relation of age, gender, and bone mass to circulating sclerostin levels in women and men," *Journal of bone and mineral research*, vol. 26, no. 2, pp. 373–379, 2011.

- [59] M. M. Roforth, K. Fujita, U. I. McGregor, S. Kirmani, L. K. McCready, J. M. Peterson, M. T. Drake, D. G. Monroe, and S. Khosla, “Effects of age on bone mrna levels of sclerostin and other genes relevant to bone metabolism in humans,” *Bone*, vol. 59, pp. 1–6, 2014.
- [60] H. Z. Ke, W. G. Richards, X. Li, and M. S. Ominsky, “Sclerostin and dickkopf-1 as therapeutic targets in bone diseases,” *Endocrine reviews*, vol. 33, no. 5, pp. 747–783, 2012.
- [61] N. Kusu, J. Laurikkala, M. Imanishi, H. Usui, M. Konishi, A. Miyake, I. Thesleff, and N. Itoh, “Sclerostin is a novel secreted osteoclast-derived bone morphogenetic protein antagonist with unique ligand specificity,” *Journal of Biological Chemistry*, vol. 278, no. 26, pp. 24 113–24 117, 2003.
- [62] K. Ota, P. Quint, M. Ruan, L. Pederson, J. J. Westendorf, S. Khosla, and M. J. Oursler, “Sclerostin is expressed in osteoclasts from aged mice and reduces osteoclast-mediated stimulation of mineralization,” *Journal of cellular biochemistry*, vol. 114, no. 8, pp. 1901–1907, 2013.
- [63] D. Zhu, N. C. W. Mackenzie, J. L. Millan, C. Farquharson, and V. E. MacRae, “The appearance and modulation of osteocyte marker expression during calcification of vascular smooth muscle cells,” *PloS one*, vol. 6, no. 5, p. e19595, 2011.
- [64] S. Scheiner, P. Pivonka, D. Smith, C. Dunstan, and C. Hellmich, “Mathematical modeling of postmenopausal osteoporosis and its treatment by the anti-catabolic drug denosumab,” *International journal for numerical methods in biomedical engineering*, vol. 30, no. 1, pp. 1–27, 2014.
- [65] B. C. NORDIN, A. G. NEED, B. E. CHATTERTON, M. HOROWITZ, and H. A. MORRIS, “The relative contributions of age and years since menopause to postmenopausal bone loss,” *The Journal of Clinical Endocrinology & Metabolism*, vol. 70, no. 1, pp. 83–88, 1990.
- [66] G. Eghbali-Fatourehchi, S. Khosla, A. Sanyal, W. J. Boyle, D. L. Lacey, B. L. Riggs *et al.*, “Role of rank ligand in mediating increased bone resorption in early postmenopausal women,” *The Journal of clinical investigation*, vol. 111, no. 8, pp. 1221–1230, 2003.
- [67] D. D. Marathe, A. Marathe, and D. E. Mager, “Integrated model for denosumab and ibandronate pharmacodynamics in postmenopausal women,” *Biopharmaceutics & drug disposition*, vol. 32, no. 8, pp. 471–481, 2011.
- [68] J. Martínez-Reina, J. L. Calvo-Gallego, and P. Pivonka, “Are drug holidays a safe option in treatment of osteoporosis?—insights from an in silico mechanistic pk–pd model of denosumab treatment of postmenopausal osteoporosis,” *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 113, p. 104140, 2021.
- [69] D. Padhi, G. Jang, B. Stouch, L. Fang, and E. Posvar, “Single-dose, placebo-controlled, randomized study of amg 785, a sclerostin monoclonal antibody,” *Journal of bone and mineral research*, vol. 26, no. 1, pp. 19–26, 2011.
- [70] M. R. McClung, J. P. Brown, A. Diez-Perez, H. Resch, J. Caminis, P. Meisner, M. A. Bolognese, S. Goemaere, H. G. Bone, J. R. Zanchetta *et al.*, “Effects of 24 months of treatment with romosozumab followed by 12 months of denosumab or placebo in postmenopausal women with low bone mineral density: a randomized, double-blind, phase 2, parallel group study,” *Journal of Bone and Mineral Research*, vol. 33, no. 8, pp. 1397–1406, 2018.
- [71] “Matlab guide: Code view de app designer,” https://es.mathworks.com/help/matlab/creating_guis/app-designer-code-generation.html, 2015, *accedido* 14 – 07 – 2023.
- [72] “Matlab guide: Componentes de app designer,” https://es.mathworks.com/help/matlab/creating_guis/choose-components-for-your-app-designer-app.html, 2015, *accedido* 14 – 07 – 2023.

- [73] R. Orueta and S. Gómez-Caro, “Interpretación de la densitometría ósea,” *SEMERGEN-Medicina de familia*, vol. 36, no. 1, pp. 27–30, 2010.
- [74] L. Warming, C. Hassager, and C. Christiansen, “Changes in bone mineral density with age in men and women: a longitudinal study,” *Osteoporosis International*, vol. 13, pp. 105–112, 2002.
- [75] “Recomendaciones osteoporosis,” https://www.ser.es/wp-content/uploads/2018/03/Recomendaciones_OP_DEF.pdf, 2015, accedido 14-07-2023.
- [76] “Qué es el machine learning,” <https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico>, 2015, accedido 14-07-2023.
- [77] “¿sabes en qué se diferencian las redes neuronales del deep learning?” <https://blogthinkbig.com/redes-neuronales-deep-learning>, 2015, accedido 14-07-2023.
- [78] “La neurona artiicial,” <https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes_neuronales/curso - glisa - redes_neuronales - html/x38.html>, 2015, accedido 14 - 07 - 2023.
- [79] “Redes neuronales,” <https://sites.google.com/site/juanmarcillo2014/contact-me>, 2015, accedido 14-07-2023.
- [80] “Inteligencia artificial y tipos de redes neuronales,” <https://inteligencia-artificial.dev/tipos-redes-neuronales/>, 2015, accedido 14-07-2023.
- [81] “Clasificación de las redes neuronales,” <https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/>, 2017, accedido 14-07-2023.
- [82] “Estructura de una red neuonal densa,” <https://www.researchgate.net/figure/Estructura-Red-neuronal-Densa-Cuenta-con-la-primera-capade-entrada-dos-capas-ocultas_{fig4}343450066>, 2015, accedido 14 - 07 - 2023.
- [83] “Matlab guide: Neural network toolbox,” <http://cda.psych.uiuc.edu/matlab_{pdf}/nnet.pdf>, 2015, accedido 14 - 07 - 2023.
- [84] “Redes neuronales,” <https://www.researchgate.net/figure/Figura-3-Red-Neuronal-Feedforward_{fig3}269466094>, 2015, accedido 14 - 07 - 2023.
- [85] “Aprendizaje redes neuronales,” <https://lamaquinaoraculo.com/computacion/aprendizaje-de-redes-neuronales/:text=El2015>, accedido 14-07-2023.
- [86] “Aplicaciones de las redes neuronales,” <https://enzyme.biz/blog/redes-neuronales-que-son-y-aplicaciones>, 2015, accedido 14-07-2023.
- [87] “Mathworks: ¿que es una red neuronal?” <https://es.mathworks.com/discovery/neural-network.html>, 2015, accedido 14-07-2023.
- [88] “Researchgate,” <https://www.researchgate.net/figure/Frosts-Mechanostat-Theory-Adaptation-of-bone-to-mechanical-stimuli_{fig1}27529980>, 2015, accedido 14 - 07 - 2023.