



Universidad de Jaén

Escuela Politécnica Superior
de Jaén

TRABAJO FIN DE GRADO

**SERIOUS GAME PARA EL
APRENDIZAJE DE LAS TÉCNICAS
DE ELECTROTERAPIA
ORIENTADA A ALUMNOS DEL
GRADO EN FISIOTERAPIA**

Alumno

Pedro Cátedra Torres

Tutores

Antonio Jesús Rueda Ruiz

(Departamento de Informática)

Carmen Martínez Cruz

(Departamento de Informática)

Septiembre, 2019

(Página intencionalmente en blanco)



Universidad de Jaén

Departamento de Informática

Don Antonio Jesús Rueda Ruiz, Doña Carmen Martínez Cruz, tutores del Trabajo Fin de Grado titulado: '**Serious game para el aprendizaje de las técnicas de electroterapia orientada a alumnos del Grado en Fisioterapia**', que presenta Pedro Cátedra Torres, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2019

Pedro Cátedra Torres

Antonio Jesús Rueda Ruiz
Carmen Martínez Cruz

(Página intencionalmente en blanco)

Agradecimientos

A mi padre Pedro, por mostrarme el valor de la humildad.

A mi madre Ana, por enseñarme el valor del esfuerzo.

A mi hermano Iván, por no dejar que saliese del camino.

A mis abuelos, a mis tíos y a mis primos por la insistencia y el cariño.

A mi novia Claudia, por el apoyo.

Agradecer también a Juan José por la oportunidad.

Gracias Antonio y Carmen por los conocimientos adquiridos.

Gracias Ángeles y Cati por vuestro tiempo.

Fue un placer trabajar con ustedes.

FICHA DEL TRABAJO FIN DE TÍTULO	
Titulación	Ingeniería informática
Modalidad	Proyecto de ingeniería
Especialidad (solo TFG)	Tecnologías de la información y comunicación
Mención (solo TFG)	Elija un elemento.
Idioma	Español
Tipo	TFG
TFT en equipo	No
Autor/a	Pedro Cátedra Torres
Fecha de asignación	Marzo de 2019
Descripción corta	Este trabajo de fin de grado pretende continuar un trabajo anterior en el que se diseñó una aplicación iOS para facilitar el estudio de las técnicas de electroterapia, orientada a alumnos del Grado en Fisioterapia. El aprendizaje de estas técnicas plantea dificultades a los alumnos debido a la aridez de la materia y al elevado número de corrientes y parámetros físicos que es necesario manejar. En este TFG queremos introducir la gamificación para motivar al alumno y evaluar de manera precisa su dominio de la asignatura. El serious game que proponemos simulará un tratamiento real de electroterapia, describiendo los síntomas de un paciente que el alumno tendrá que tratar eligiendo un tipo de corriente, configurando los parámetros de dicha corriente y el tiempo de tratamiento, y colocando los correspondientes electrodos en el cuerpo del paciente. Usaremos una evaluación mediante lógica difusa del tratamiento realizado por el alumno para que sea lo más realista y flexible posible, es decir, que se parezca lo máximo posible a la evaluación realizada por el profesor de la asignatura o por un experto de la materia.

NORMAS APLICADAS EN ESTE DOCUMENTO

LOCALES	
TFT-UJA:2017	Normativa de Trabajos Fin de Grado, Fin de Máster y otros Trabajos Fin de Título de la Universidad de Jaén (Normativa marco UJA aprobada en Consejo de Gobierno)
TFT-EPSJ:2017	Normativa sobre Trabajos Fin de Grado y Fin de Máster en la Escuela Politécnica Superior de Jaén (Normativa EPSJ aprobada en Junta de Escuela)
TFT-EPSJ	Criterios de evaluación y normas de estilo para TFG y TFM de la Escuela Politécnica Superior de Jaén
NACIONALES E INTERNACIONALES	
ISO 2145:1978	Documentación - Numeración de divisiones y subdivisiones en documentos escritos
UNE 50132:1994	Traducción de la ISO 2145
APA 6ª edición	Estilo de referencias y citas de APA (American Psychological Association)

NORMAS UTILIZADAS COMO BASE O REFERENCIA

NACIONALES	
UNE 157001:2014	Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico
UNE 157801:2007	Criterios generales para la elaboración de proyectos de sistemas de información
<i>Estas normas se han utilizado como base o referencia para la inclusión de algunos contenidos y definiciones sobre elaboración de proyectos, entendiendo como proyecto la documentación consensuada entre una empresa y un cliente, que da lugar al perfeccionamiento de un contrato para la elaboración de una obra o la prestación de un servicio. Por consiguiente, no debe esperarse la aplicación de estas normas en cuanto a la completitud de los contenidos ni a la organización de los mismos.</i>	

Contenido

1	Especificación del trabajo	13
1.1	Introducción	13
1.2	Objetivos del trabajo	14
1.3	Antecedentes y estado del arte	14
1.4	Hipótesis y restricciones	15
1.5	Tecnologías utilizadas	16
1.5.1	Blender	16
1.5.2	IOS - Swift	16
1.5.3	Xcode	16
1.5.4	FuzzyLite	16
1.6	Metodología de desarrollo de software	17
1.7	Estimación del tamaño y esfuerzo	17
1.8	Planificación temporal	18
2	Desarrollo	19
2.1	Sprint 1	20
2.1.1	Estudio de la app y código	20
2.1.2	Migración código Swift 3	20
2.1.3	Reunión	21
2.1.4	Nueva imagen y zonas	21
2.1.5	Nuevos iconos de familia de corrientes	22
2.2	Sprint 2	23
2.2.1	Sistema de identificación de zonas	23
2.2.2	Nuevos electrodos	24
2.2.3	Nueva opción de modulación	24
2.3	Sprint 3	26
2.3.1	Delimitación de las zonas con elipsoides	26
2.4	Sprint 4	29
2.4.1	Reunión	29
2.4.2	Desarrollo técnica tetrapolar	30
2.4.3	Desarrollo método segmentario	31
2.5	Sprint 5	33
2.6	Sprint 6	38
2.6.1	Mostrar mensajes de error	38
2.6.2	Reunión	39
2.6.3	Añadir botones para seleccionar el modo de juego	40
2.6.4	Nueva vista caso clínico	40
2.6.5	Archivo JSON para la lectura del caso clínico	41
2.7	Sprint 7	43
2.7.1	Desarrollo de un controlador para los modos de juego	43
2.8	Sprint 8	47
2.8.1	Reunión	47
2.8.2	Crear niveles de sensibilidad	48
2.8.3	Inclusión de un paciente	48
2.8.4	Spike para la evaluación	50
2.9	Sprint 9	51
2.9.1	Añadir solución al archivo JSON	51

2.9.2	Creación del archivo Fuzzylite.....	52
2.10	Sprint 10.....	56
2.10.1	Desarrollo evaluación difusa	56
2.11	Sprint 11.....	58
2.11.1	Desarrollo evaluación no difusa	58
2.11.2	Nueva vista evaluación.....	60
2.12	Sprint 12.....	61
2.12.1	Desarrollo para el cálculo de la nota final.....	61
2.12.2	Añadir nuevos casos clínicos	62
2.13	Pruebas finales	63
3	Experimentación, resultados y discusión	64
3.1	Experimentaciones y pruebas.....	64
3.2	Resultados y discusión	64
4	Manual de usuario.....	65
4.1	Storyboard	65
4.2	Vista selección modo de juego	66
4.3	Modo aprendizaje	67
4.3.1	Vista configuración del tratamiento	67
4.3.2	Vista selección de zona.....	68
4.3.3	Vista colocación de electrodos	70
4.3.4	Selección familia corriente dispositivo.....	71
4.3.5	Vista selección corriente dispositivo.....	72
4.3.6	Vista configuración dispositivo	73
4.3.7	Vista inicio dispositivo.....	74
4.4	Modo evaluación	76
4.4.1	Vista caso clínico.....	76
4.4.2	Vista colocación de electrodos	77
4.4.3	Vista selección familia corriente dispositivo	78
4.4.4	Vista selección corriente dispositivo.....	79
4.4.5	Vista configuración dispositivo	80
4.4.6	Vista inicio dispositivo.....	81
4.4.7	Vista evaluación	82
5	Conclusiones y trabajos futuros	83
6	Apéndices	84
6.1	Zonas corporales	84
6.2	Tablas de corrientes, valores y mensajes.....	85
6.3	Cuestionario de opinión	86
6.4	Resultados cuestionario de opinión	88
6.5	Guía original del Trabajo Fin de Título.....	89
7	Bibliografía	91

Índice de ilustraciones

Ilustración 1.1	16
Ilustración 2.1	20
Ilustración 2.2	21
Ilustración 2.3	22
Ilustración 2.4	23
Ilustración 2.5	25
Ilustración 2.6	26
Ilustración 2.7	27
Ilustración 2.8	27
Ilustración 2.9	28
Ilustración 2.10	29
Ilustración 2.11	30
Ilustración 2.12	30
Ilustración 2.13	31
Ilustración 2.14	32
Ilustración 2.15	33
Ilustración 2.16	34
Ilustración 2.17	34
Ilustración 2.18	34
Ilustración 2.19 Ilustración 2.20	35
Ilustración 2.21	36
Ilustración 2.22 Ilustración 2.23	38
Ilustración 2.24 Ilustración 2.25	39
Ilustración 2.26	40
Ilustración 2.27	40
Ilustración 2.28	41
Ilustración 2.29	42
Ilustración 2.30	45
Ilustración 2.31	48
Ilustración 2.32 Ilustración 2.33	49
Ilustración 2.34 Ilustración 2.35	49
Ilustración 2.36	49
Ilustración 2.37	51
Ilustración 2.38	53
Ilustración 2.39	53
Ilustración 2.40	54
Ilustración 2.41	60
Ilustración 4.1	65
Ilustración 4.2	66
Ilustración 4.3	67
Ilustración 4.4	68
Ilustración 4.5	69
Ilustración 4.7	71

Ilustración 4.8	72
Ilustración 4.9	73
Ilustración 4.10	74
Ilustración 4.11	75
Ilustración 4.12	76
Ilustración 4.13	77
Ilustración 4.14	78
Ilustración 4.15	79
Ilustración 4.16	80
Ilustración 4.17	81
Ilustración 4.19	82

Índice de tablas

Tabla 1.1	18
Tabla 2.1	20
Tabla 2.2	23
Tabla 2.3	24
Tabla 2.4	26
Tabla 2.5	29
Tabla 2.6	33
Tabla 2.7	38
Tabla 2.8	43
Tabla 2.9	44
Tabla 2.10	46
Tabla 2.11	47
Tabla 2.12	51
Tabla 2.13	56
Tabla 2.14	58
Tabla 2.15	61
Tabla 2.16	63

1 ESPECIFICACIÓN DEL TRABAJO

1.1 Introducción

La electroterapia consiste en el tratamiento de dolencias físicas a partir de la aplicación de energía electromagnética al organismo. La colocación de los electrodos por parte del fisioterapeuta requiere de conocimiento sobre anatomía, los factores causantes de la dolencia, así como el tratamiento recomendado para el paciente.

Actualmente la docencia de esta técnica está basado en ejemplos prácticos de elementos físicos sobre un paciente también físico. La enseñanza está basada en ejemplos visuales utilizando esquemas, imágenes...

Debido a esto hace unos años se llevó a cabo un proyecto de innovación docente para el desarrollo de una aplicación destinada a la práctica de la electroterapia.

El proyecto no se llegó a terminar y años más tarde se reabre de nuevo dando un paso más, un nuevo desarrollo que permita al alumno autoevaluarse.

Los serious games son “juegos formativos” en el que el entretenimiento pasa a segundo plano y el objetivo principal es de otra índole, en este caso, académico.

1.1.1 Electroterapia

En el tratamiento de la electroterapia hay dos fases bien diferenciadas:

- Colocación de electrodos: En el que se colocan un número y un tipo de electrodos con una distribución determinada en la zona de la dolencia.
- Configuración de la máquina de electroterapia: En la que se configura la corriente así como cada uno de sus parámetros. Además es de vital importancia el tiempo de tratamiento así como la intensidad proporcionada al paciente.

1.2 Objetivos del trabajo

1. Terminar y mejorar la versión del modo aprendizaje actual.
2. Implantar el uso y contrastar la mejora del aprendizaje en la materia de electroterapia utilizando la aplicación móvil.
3. Desarrollar una nueva versión de esta aplicación que permita, de un lado, la autoevaluación por parte del alumnado y fomente su aprendizaje autónomo y de otro, la evaluación objetiva por parte del docente.
4. Validar la efectividad de esta nueva versión de la aplicación como instrumento de evaluación.

Para el cumplimiento de estos objetivos habrá que llevar a cabo reuniones interdisciplinares con las clientes (las profesoras de fisioterapia) y un desarrollo del software ágil centrada en el usuario.

Se realizarán pruebas de usabilidad y satisfacción por parte de los usuarios e incorporación de posibles mejoras a la aplicación.

1.3 Antecedentes y estado del arte

Este proyecto parte de una petición de dos profesoras del grado de Fisioterapia de la Universidad de Jaén al departamento de informática hace unos años, con la intención de comenzar un proyecto de una herramienta informática para facilitar y mejorar la docencia por parte de los docentes y el estudio o entrenamiento por parte de los alumnos. Este TFG se encuadra dentro de un proyecto de innovación docente dirigido por Juan José Delgado, se inició con un TFG [4] de un alumno (Antonio Espinosa) del grado de ingeniería informática, en el que el profesor Antonio Rueda colaboró activamente. Debido a la coyuntura en ese periodo de la Universidad con los productos Apple a través de varias licencias y al auge en el desarrollo de las tecnologías iOS se optó por realizar una aplicación móvil para dispositivos iOS.

La parte del aprendizaje se presentó en un congreso internacional [3].

Si hacemos una búsqueda sobre aplicaciones relacionadas con la electroterapia podemos encontrar algunas que sirven como herramienta a profesionales y pacientes pero ninguna, que tenga la finalidad de este proyecto.

Aunque un alumno puede usar estas aplicaciones para practicar e incluso realizar algunos test con preguntas relacionadas con la electroterapia, no recibirá una

evaluación tan precisa como la que realiza nuestra aplicación. Una aplicación destinada al aprendizaje y evaluación de la técnica de la electroterapia es algo único con un enorme potencial académico.

1.4 Hipótesis y restricciones

El TFG se define como una asignatura de 12 créditos, lo que supone que la duración total del proyecto sería de 300 horas, incluyendo todas las etapas del ciclo de vida, con la excepción del mantenimiento. Sin embargo, en mi caso, se debe añadir la realización de unas prácticas Ícaro durante 6 meses, lo que equivalen a 600 horas adicionales. En total 900 horas.

Por consiguiente y aunque el número de horas sea considerable, la principal restricción aplicable es la limitación de la duración del trabajo.

1.5 Tecnologías utilizadas

1.5.1 Blender

Blender es un programa informático multiplataforma usado para modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales.

Lo utilizamos para crear los elipsoides y para la obtención de sus parámetros.

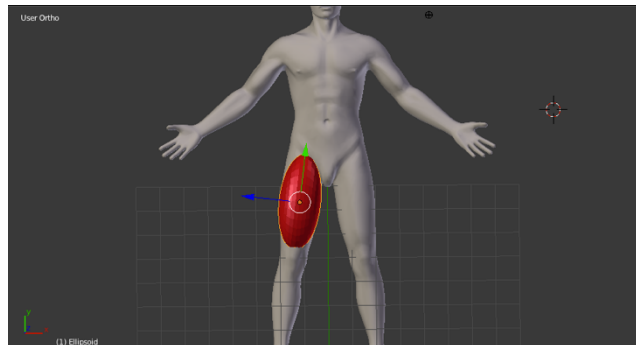


Ilustración 1.1

1.5.2 IOS - Swift

Swift es un lenguaje de programación creado por Apple enfocado al desarrollo de aplicaciones IOS y macOS. Swift es un lenguaje muy potente, simple y moderno. Con una documentación cada vez mayor y una comunidad activa.

1.5.3 Xcode

Es un entorno de desarrollo integrado creado por Apple destinado a la creación de software para dispositivos Apple: iPad, macOS, iPhone...

Xcode es fácil de usar y cuenta con varios simuladores de cada dispositivo para poder hacer pruebas sin tener que instalar el software en un dispositivo externo.

1.5.4 FuzzyLite

Es una librería open-source externa programa en C++ para el control de lógica difusa. Nos servirá para realizar una parte de la evaluación del proceso de tratamiento.

1.6 Metodología de desarrollo de software

Comencé en este proyecto gracias a unas prácticas externas Ícaro con una duración de 6 meses. Hay mucho que hacer y no se tiene claro la forma en que se van a realizar las tareas para poder alcanzar el resultado deseado, por tanto, se decide trabajar en SCRUM y alcanzar el mejor resultado posible en la duración de mis prácticas sin fechas límites ni estimaciones de ningún tipo.

Cada sprint tendrá una duración de 2 semanas en la que me reuniré con las clientes al final del sprint para mostrar los avances de la aplicación y decidir las tareas que se realizarán en el siguiente sprint según la prioridad de realización.

Cuando se crea una nueva tarea hay que estimarla con una puntuación siguiendo un equilibrio entre complejidad y carga de trabajo. Nosotros usaremos una serie parecida a la de Fibonacci: 0, ½, 1, 2, 3, 5, 8, 13, 20, 50, 100, infinito. Acordamos que una tarea no puede superar una estimación de 8 puntos, en el caso de superarla, esa tarea debe ser subdividida en dos o más tareas

1.7 Estimación del tamaño y esfuerzo

Ya que el presente proyecto es un TFG, no existen restricciones de tipo económico, sino de tipo temporal (un número aproximado de horas). Por consiguiente, los cálculos de tamaño del proyecto están supeditados al tiempo disponible. En cuanto al esfuerzo, se dispone de varias personas

Suponiendo un sueldo mensual a media jornada de un desarrollador rondará los 1000€ aproximadamente. La realización de este proyecto se llevó a cabo en 6 meses. El coste total de este proyecto es de aproximadamente 6000€. Hay que añadir el coste del tiempo de las clientes y las pruebas con los alumnos.

1.8 Planificación temporal

TAREA	FECHA
Sprint 1	1 - 15 de Febrero
Sprint 2	18 de Febrero - 1 de Marzo
Sprint 3	4 - 15 de Marzo
Sprint 4	18 - 29 de Marzo
Sprint 5	1 - 12 de Abril
Sprint 6	15 - 26 de Abril
Sprint 7	29 de Abril - 10 de Mayo
Sprint 8	13 - 24 de Mayo
Sprint 9	27 de Mayo - 7 de Junio
Sprint 10	10 - 21 de Junio
Sprint 11	24 de Junio - 5 de Julio
Sprint 12	8 - 19 de Julio
Corrección de bugs	22 - 31 de Julio

Tabla 1.1

Hay que aclarar que la planificación temporal se realiza con el tiempo de mis prácticas Ícaro. Los sprint pertenecientes al TFG, son a partir del octavo (13 de Mayo).

2 DESARROLLO

Como trabajaremos en SCRUM no hay requisitos iniciales ya que estos son cambiantes y poco definidos. Los requisitos de cada tarea serán definidos en la reunión correspondiente con el siguiente sprint. Las reuniones con las que contaremos son las siguientes:

- Refinamiento: en la que se deciden junto con las product owners las tareas que se deben crear y añadir al sprint backlog. Además se deciden aquellas tareas que entran para el siguiente sprint.
- Refinamiento interno: el equipo de desarrollo estima las tareas. Debido a temas de horarios, esta función la hago yo generalmente.
- Demo: se muestra a las product owners el resultado del sprint.

Las tareas que entran al sprint dependen de la prioridad de las mismas.

Contaremos con un sprint backlog en el que se encontrarán todas las tareas pendientes que no entran en un sprint debido a cuestiones de prioridad o tiempo.

Los miembros del equipo de desarrollo de este proyecto son los siguientes:

- Pedro Cátedra Torres: Developer y Tester.
- Carmen Martínez Cruz: Developer.
- Antonio Jesús Rueda: Chief architect.
- Juan José Delgado: Scrum master.
- Ángeles Díaz: Product owner.
- María Catalina Osuna: Product owner.

A partir de aquí se nombrará a cada miembro con su papel en el equipo de desarrollo.

2.1 Sprint 1

TAREA	ESTIMACIÓN	VISTA
Estudio de la app y código	2	
Migración código Swift 3.0	2	
Nueva imagen y zonas	2	Selección zona
Cambios menores familias de corrientes	2	Selección familia corriente dispositivo

Tabla 2.1

2.1.1 Estudio de la app y código

Debido a que me encuentro con un proyecto ya iniciado necesitaba de un tiempo para familiarizarme con la aplicación, estudiando toda su funcionalidad y aprendiendo todo lo relativo de la electroterapia que fuese necesario para el desarrollo de la aplicación.

Además, revisé el código estudiando cada uno de los archivos y cómo estaba enlazado con cada una de las vistas.

2.1.2 Migración código Swift 3

Una vez familiarizado con la aplicación, comencé a migrar el código de la versión Swift 2.0 a Swift 3.0. Esto se debe a que el proyecto comenzó hace varios años y quedó un poco estancado. La versión del código estaba desactualizada así que antes de comenzar con el desarrollo había que cambiar aquellas líneas de código cuya sintaxis era errónea.

Aquí un ejemplo de la migración, las líneas en rojo son las desactualizadas y las verdes las líneas actualizadas.

```

87 - self.electrodeType = ElectrodeType(rawValue: aDecoder.decodeInteger(forKey: "shape"))!
88 - self.polarity = ElectrodePolarity(rawValue: aDecoder.decodeInteger(forKey: "polarity"))!
89 - self.selected = aDecoder.decodeBool(forKey: "selected")
90 - self.blinking = aDecoder.decodeBool(forKey: "blinking")

87 + self.electrodeType = ElectrodeType(rawValue: aDecoder.decodeInteger(forKey: "shape"))!
88 + self.polarity = ElectrodePolarity(rawValue: aDecoder.decodeInteger(forKey: "polarity"))!
89 + self.selected = aDecoder.decodeBool(forKey: "selected")
90 + self.blinking = aDecoder.decodeBool(forKey: "blinking")

```

Ilustración 2.1

2.1.3 Reunión

Tras estas dos primeras tareas, me reuní con el equipo. Nos encontrábamos todos los integrantes del equipo (excepto Carmen Martínez Cruz que se incorporó más tarde). En ella, me presentaron a las product owner (Ángeles y María) y me explicaron los **objetivos** del proyecto:

5. Terminar y mejorar la versión del modo aprendizaje actual.
6. Implantar el uso y contrastar la mejora del aprendizaje en la materia de electroterapia utilizando la aplicación.
7. Desarrollar una nueva versión de esta aplicación que permita, de un lado, la autoevaluación por parte del alumnado y fomente su aprendizaje autónomo y de otro, la evaluación objetiva por parte del docente.
8. Validar la efectividad de esta nueva versión de la aplicación como instrumento de evaluación.

2.1.4 Nueva imagen y zonas

En la reunión anterior las clientas me ofrecieron una lista con todas las zonas del cuerpo humano que podrán ser tratadas en la aplicación (anexo 1).

El chief architect me proporcionó unas imágenes de un modelo del cuerpo humano más moderno para incluirlo en la vista 'selección de zona'. También había que incluir un botón por cada zona nueva además de darle un estilo acorde al de la imagen (Ilustración 2.2).

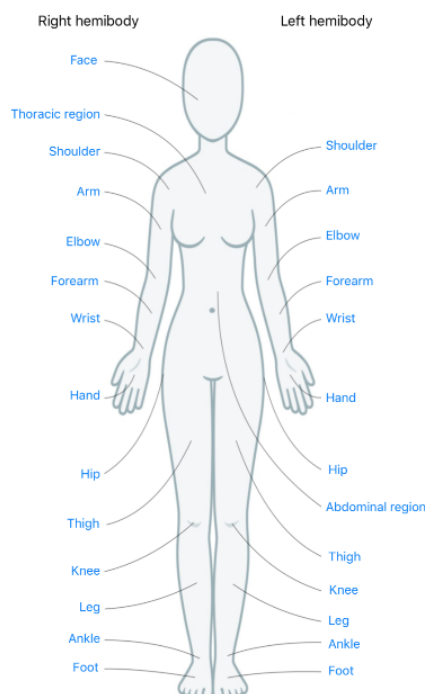


Ilustración 2.2

2.1.5 Nuevos iconos de familia de corrientes

Para la vista ‘selección familia de corriente dispositivo’ en la que aparecen las familias de corrientes se renuevan algunos iconos antiguos por unos más actualizados además de cambios menores en los nombres.

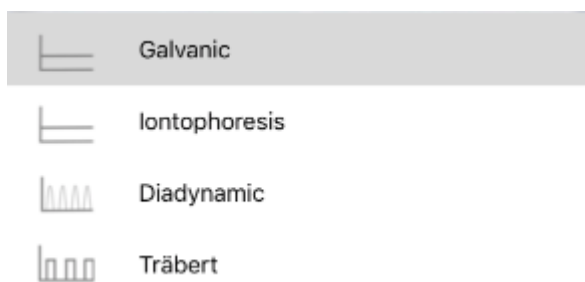


Ilustración 2.3

2.2 Sprint 2

TAREA	ESTIMACIÓN	VISTA
Sistema de identificación de zonas	2	Selección de zona
Nuevos electrodos	1	Colocación de electrodos
Nueva opción de modulación	5	Configuración dispositivo

Tabla 2.2

2.2.1 Sistema de identificación de zonas

Para poder diferenciar cada una de las zonas del cuerpo se necesita de un sistema para identificarlas. Inicialmente a cada zona del cuerpo seleccionable se le asignaba un número que las identificaba. Aquí varios ejemplos de como estaba en el código:

Brazo frontal izquierdo -> 300

Cara -> 1121

Muslo frontal derecho -> 2210

Este sistema de identificación me pareció altamente ininteligible, por lo que decidí cambiarlo. En lugar de asignarle un número mejor tener un enumerado. De esta forma el código queda más limpio.

```
enum HumanBodyZone : Int {  
    case CaraF = 0, RegionCervicalT, RegionDorsalT, RegionToracicaF, TrapecioTI,  
        TrapecioTD  
    case HombroFI, HombroFD, RegionLumbarT, BrazoFI, BrazoFD, BrazoTI, BrazoTD  
    case CodoFI, CodoFD, CodoTI, CodoTD, AntebrazoFI, AntebrazoFD, AntebrazoTI,  
        AntebrazoTD  
    case RegionAbdominalF, MuñecaFI, MuñecaFD, MuñecaTI, MuñecaTD  
    case ManoFI, ManoFD, ManoTI, ManoTD, RegionLumbarBajaT, RegionSacraT, CaderaFI,  
        CaderaFD  
    case GluteoTI, GluteoTD, MusloFI, MusloFD, MusloTI, MusloTD, RodillaFI, RodillaFD  
    case PiernaFI, PiernaFD, RegionGemelarTI, RegionGemelarTD  
    case TobilloFI, TobilloFD, TobilloTI, TobilloTD, PieFI, PieFD  
}
```

Ilustración 2.4

F: Frontal T: Trasero D: derecha I: izquierda

2.2.2 Nuevos electrodos

Se decide cambiar los electrodos de la vista 'colocación de electrodos'. Inicialmente se cuentan con tres tipos de electrodos:

1. Rectangular 90x50 mm
2. Cuadrado 50x50 mm
3. Redondo 50 mm
4. Redondo 32 mm

Se cambian por los siguientes:

1. Rectangular 120x80 mm
2. Rectangular 60x80 mm
3. Rectangular 60x40 mm
4. Redondo 32 mm

Los cambios son menores, simplemente hay que cambiar las dimensiones de los que ya existían además de sus imágenes.

2.2.3 Nueva opción de modulación

Las clientes piden que en la vista 'configuración dispositivo' aparezca una nueva opción llamada 'modulación'. Esta opción no debe aparecer en todas las familias de corrientes (ver anexo 2) y sus valores dependen también de la familia. Requisitos:

Familia de corriente	Corriente	Opciones
TENS	Asimétrico	Amplitud, frecuencia, No
TENS	Simétrico	Trenes, No
Interferencial	Bipolar	Espectro, No
Interferencial	Tetrapolar	Espectro, No
Interferencial	Estimulación rusa	Trenes, No

Tabla 2.3

Para desarrollar el comportamiento de un apartado en el dispositivo y hacerlo de una manera genérica existe un protocolo 'ConfigParameter'. Un protocolo es una

especie de interfaz en Swift en la que también se pueden especificar propiedades que deben ser implementadas.

Hay que crear una clase que implementa este protocolo. Recibiendo en el constructor el tipo de corriente podemos discriminar las opciones disponibles. Las funciones de incremento y decremento están directamente relacionadas con los botones 'dosage' de la vista 'configuración dispositivo'.

```
/** Configuration parameter protocol */  
protocol ConfigParameter {  
    var parameterLabel: String { get }  
  
    var valueLabel: String { get }  
  
    var isReadOnly: Bool { get }  
  
    func increase()  
    func decrease()  
}
```

Ilustración 2.5

En este apartado de configuración estas funciones simplemente avanzan o retroceden la posición en el array de las opciones disponibles.

En un apartado con opciones numéricas como por ejemplo la frecuencia, estas funciones aumentan o disminuyen el valor numérico.

2.3 Sprint 3

TAREA	ESTIMACIÓN	VISTA
Delimitación de zonas con elipsoides	8	Colocación de electrodos

Tabla 2.4

2.3.1 Delimitación de las zonas con elipsoides

Inicialmente la delimitación de zonas se realizaba mediante planos que intersectaban y aislaban cada una de las zonas del cuerpo que se pueden seleccionar. Veamos un ejemplo:

Para aislar la zona 'CodoTI' se requería de 5 planos. Cuando se coloca un electrodo sabemos las coordenadas del punto del espacio donde se coloca, por tanto, para saber si dicho punto pertenece a la zona escogida en la vista 'selección de zona', debe cumplir unas restricciones. Cada restricción es un operador (mayor o menor) y un 'plano' que se define con una coordenada en un eje (x/y/z). Un punto pertenece a la zona si cada coordenada x/y/z del punto seleccionado en el espacio cumple la condición del operador con la coordenada del plano. Por ejemplo para la primera restricción, la coordenada 'y' del punto seleccionado debe ser mayor que 11.6.

```
restrictions.append(Restriction(plane: yPlane, mid: >, value: 11.60))
restrictions.append(Restriction(plane: yPlane, mid: <, value: 13.52))
restrictions.append(Restriction(plane: xPlane, mid: >, value: 1.80))
restrictions.append(Restriction(plane: xPlane, mid: <, value: 3.65))
restrictions.append(Restriction(plane: zPlane, mid: <, value: -0.8))
```

Ilustración 2.6

Tras una reunión con el chief architect se decide cambiar este sistema por una encapsulación con elipsoides. Se eliminan los planos y se pretende aislar la zona del cuerpo mediante un elipsoide. Para ello usé el programa gráfico Blender y definí un elipsoide para cada una de las zonas del cuerpo seleccionables. En las imágenes se pueden ver las zonas 'MusloFD' y 'HombroFD' respectivamente.'

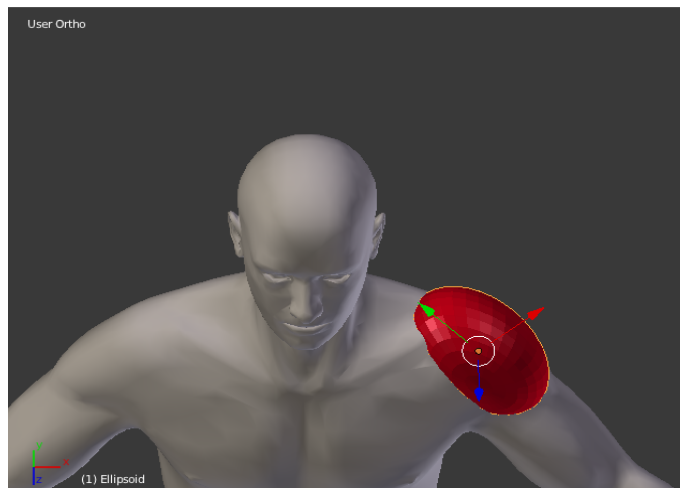


Ilustración 2.7

El eje Y (verde) debe estar alineado con el eje longitudinal de la zona.

El eje X (rojo) debe ser la normal al plano de la zona.

Una vez tengamos la zona perfectamente cubierta necesitamos obtener cuatro parámetros del elipsoide para poder saber si un punto se encuentra dentro o fuera de él:

1. Las coordenadas del punto central.
2. El vector VY (verde).
3. El vector VX (rojo).
4. El vector VZ (azul).

Para obtener estos valores hice uso de un script en Python suministrado por el chief architect:

```
def ellipsoid_params():  
    object = bpy.data.objects["Ellipsoid"]  
    p = object.location  
    world3x3 = object.matrix_world.to_3x3()  
    vx = world3x3 * mathutils.Vector((1.0, 0.0, 0.0))  
    vy = world3x3 * mathutils.Vector((0.0, 1.0, 0.0))  
    vz = world3x3 * mathutils.Vector((0.0, 0.0, 1.0))  
  
    return (p, vx, vy, vz)
```

Ilustración 2.8

Para cada una de las zonas hay que almacenar estos cuatro valores como una restricción. Un punto pertenece al elipsoide si está contenido en él, para ello usamos esta función (refactorizada a Swift) definida también en el script:

```
func contains(_ q: SCNVector3) -> Bool {  
    let a = length(vx)  
    let b = length(vy)  
    let c = length(vz)  
  
    let v = q - p  
    let x = dot(v, vx) / a  
    let y = dot(v, vy) / b  
    let z = dot(v, vz) / c  
  
    return x * x / (a * a) + y * y / (b * b) + z * z / (c * c) <= 1  
}
```

Ilustración 2.9

Aunque la tarea no es excesivamente compleja tuvo una gran carga de trabajo debido a la gran cantidad de elipsoides que se tuvieron que crear, cincuenta y seis para ser exactos.

2.4 Sprint 4

TAREA	ESTIMACIÓN	VISTA
Desarrollo técnica tetrapolar	5	Colocación de electrodos
Desarrollo método segmentario	5	Colocación de electrodos

Tabla 2.5

2.4.1 Reunión

Aunque ya había tenido una toma de contacto con las product owner, las anteriores tareas habían sido definidas con el chief architect. Había que comenzar con el desarrollo del método segmentario y la técnica tetrapolar así que necesitaba obtener los requisitos para la realización de estas tareas. Esta fue la información que pude obtener de la reunión:

- Técnica tetrapolar:
 1. Deben poder colocarse cuatro electrodos del mismo tamaño en la zona a tratar.
 2. Sólo se pueden colocar en la misma zona.
 3. Dos electrodos positivos y dos negativos.
 4. Cada electrodo positivo con su respectivo electrodo negativo forman un circuito.
 5. Los circuitos deben tener colores diferentes.
 6. Ambos circuitos deben cruzarse, de tal manera que si hubiera una línea que une los dos electrodos de un circuito, éstos quedarían formando una 'X'.

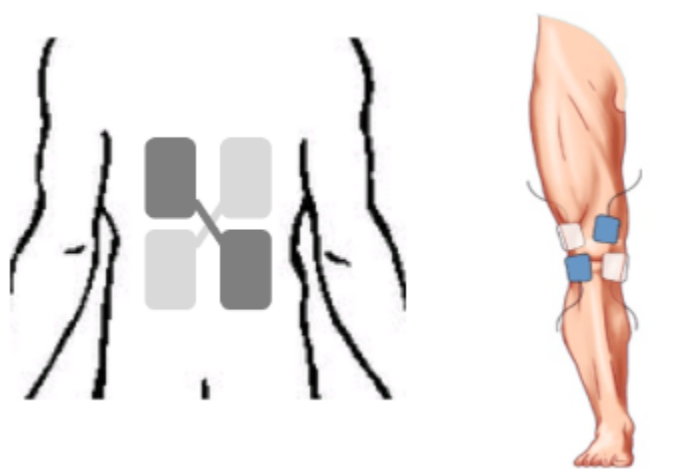


Ilustración 2.10

- Método segmentario:
 1. Deben poder colocarse dos electrodos en diferentes zonas.
 2. El electrodo positivo se coloca en la zona de la columna.
 3. El electrodo negativo en la zona a tratar.
 4. Se permite este método si la zona a tratar se encuentra en un miembro. Por ejemplo: antebrazo, mano, muslo...
 5. Se permite este método si la técnica escogida es diferente de la tetrapolar.
 6. La relación de la zona de la columna con la zona a tratar es la siguiente:
 - Cualquier zona del brazo derecho -> Cervical derecha
 - Cualquier zona del brazo izquierdo -> Cervical izquierda
 - Cualquier zona de la pierna derecha -> Lumbar derecha
 - Cualquier zona de la pierna izquierda -> Lumbar izquierda

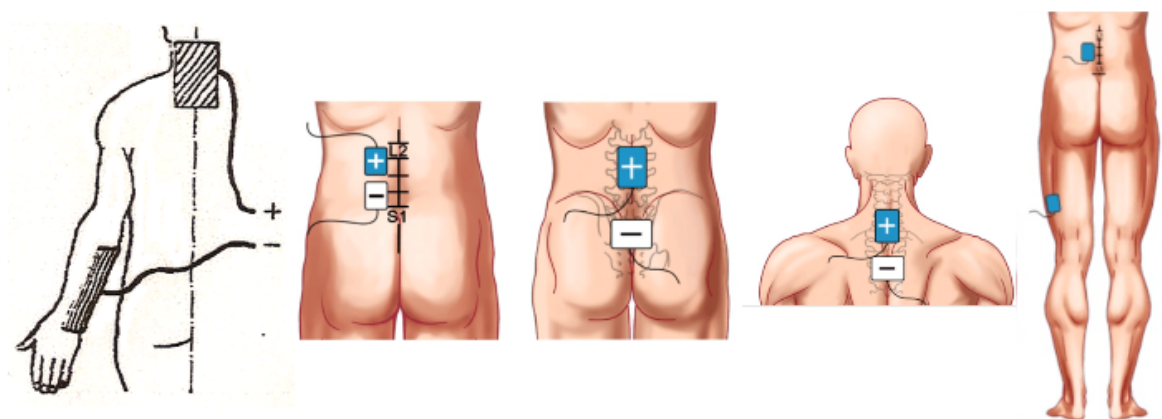


Ilustración 2.11

2.4.2 Desarrollo técnica tetrapolar

Para desarrollar esta técnica necesitamos añadir imágenes de los electrodos con un color diferente a los que ya teníamos. También, hay que añadir la opción para cambiar de circuito en el menú de cambios de electrodos de la vista 'colocación de electrodos' (Ilustración 2.12).



Ilustración 2.12

Una vez hecho esto, podemos colocar los electrodos y cambiar el circuito pero hay que comprobar que se colocan los dos circuitos y que además se cruzan. Para ello creamos dos restricciones:

- 'TwoCircuitsInTetrapolarCheck': chequea si existen dos circuitos.
- 'CrossCircuitsInTetrapolarCheck': chequea si ambos circuitos se cortan formando una 'X' [1].

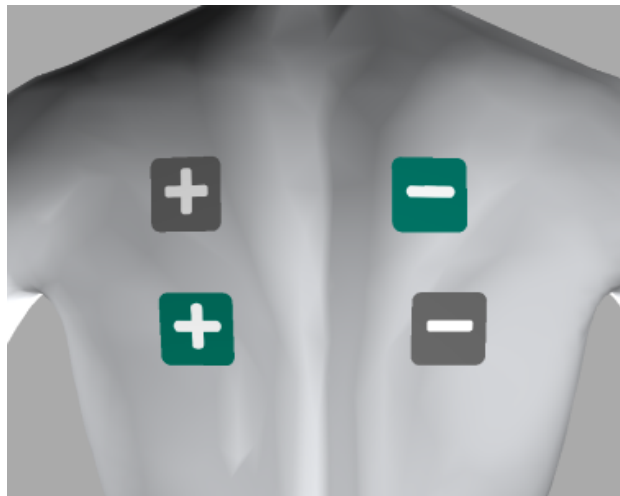


Ilustración 2.13

2.4.3 Desarrollo método segmentario

Para el desarrollo de este método hay que aceptar que se pueda tener dos zonas a evaluar, y por tanto, dos elipsoides. El segundo elipsoide (zona de la columna) vendrá determinado en función del primer elipsoide (zona a tratar).

Existe un problema al implementar este método y es que la cámara se centra en la zona a tratar y no se puede mover hacia la zona de la columna. Por tanto, hay que mover la cámara al punto medio entre la zona a tratar y la zona de la columna.

Hay que crear una restricción llamada 'SegmentalBackZoneCheck' que comprueba que el electrodo positivo se encuentra en el segundo elipsoide (zona de la columna) y el electrodo negativo en el primer elipsoide (zona a tratar).



Ilustración 2.14

2.5 Sprint 5

TAREA	ESTIMACIÓN	VISTA
Evaluar colocación de electrodos	8	Colocación de electrodos

Tabla 2.6

En el modo aprendizaje la aplicación debe guiar al usuario durante el proceso de la electroterapia. Actualmente el usuario tiene las herramientas necesarias (imágenes, modelo, opciones, etc.) para posicionar todos los electrodos posibles en la forma que determine, pero éstos deben ser evaluados y comprobar que la técnica y método escogidos en la vista ‘configuración del tratamiento’ se corresponden con la colocación de los electrodos en el modelo. Tienen que cumplir los requisitos mencionados en el sprint anterior en cuanto a la técnica tetrapolar y al método segmentario además de los siguientes nuevos requisitos:

- Técnica bipolar:
 1. Deben poder colocarse dos electrodos en diferentes zonas.
 2. Deben ser del mismo tamaño
 3. Deben tener diferente polaridad.

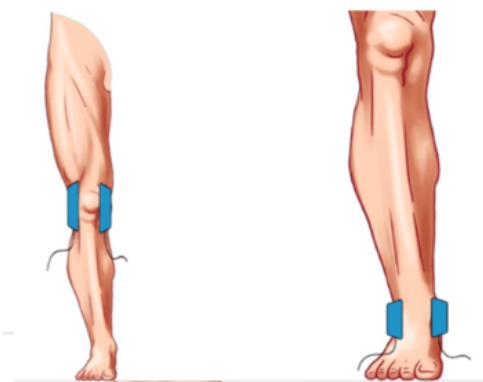


Ilustración 2.15

- Técnica monopolar:
 1. Deben poder colocarse dos electrodos en diferentes zonas.
 2. Deben tener diferente polaridad.
 3. El electrodo positivo debe ser de menor tamaño que el negativo.

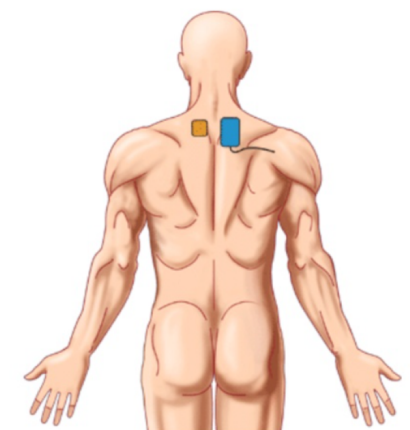
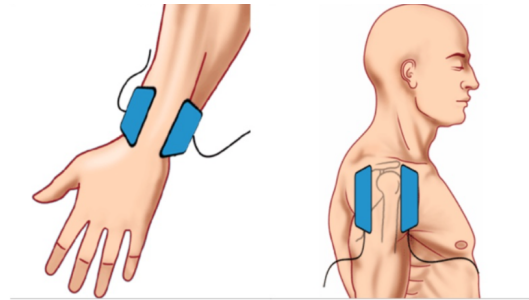
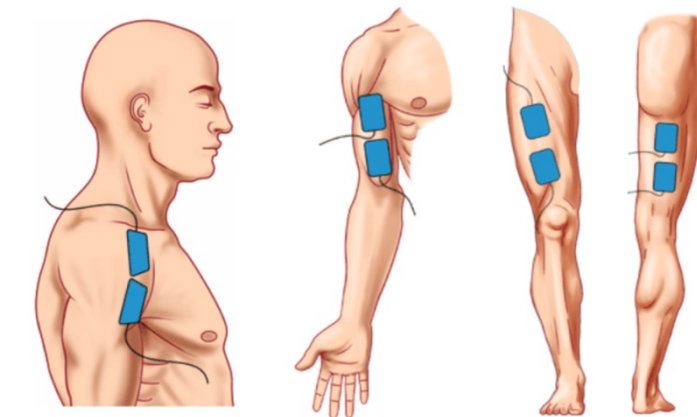


Ilustración 2.16

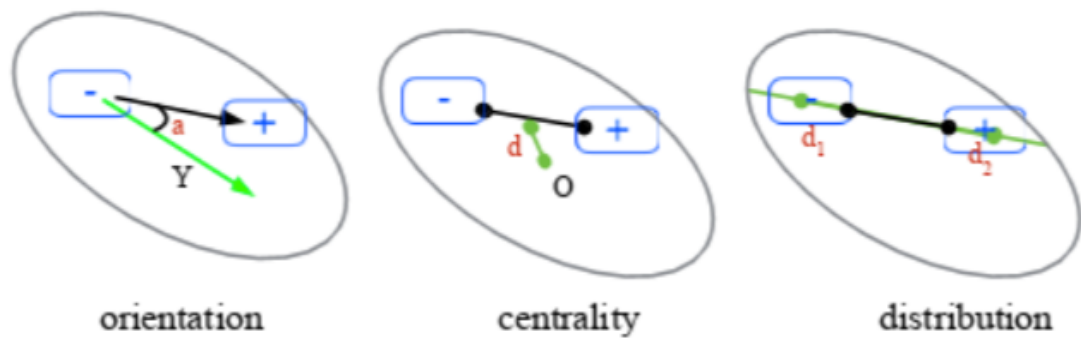
- Método transversal:
 1. Los electrodos se colocan paralelos al eje transversal de la zona.

**Ilustración 2.17**

- Método longitudinal:
 1. Los electrodos se colocan paralelos al eje longitudinal de la zona.

**Ilustración 2.18**

Además de estos requisitos el chief architect elaboró una serie de medidas que determinan la calidad de la colocación de los electrodos en la zona a tratar. Son tres:



- Orientación: Valor entre 0 y 1 indicando la orientación de los electrodos respecto al eje mayor del elipsoide (0 -> transversal, 1 -> longitudinal).
- Centralidad: Valor entre 0 y 1 indicando la distancia de los electrodos respecto al punto central de elipsoide. Ejemplo de electrodos mal centrados en la zona (Ilustración 2.19).
- Distribución: Valor entre 0 y 1 indicando la distancia de separación entre los electrodos. Ejemplo de electrodos mal distribuidos en la zona (Ilustración 2.20).



Ilustración 2.19



Ilustración 2.20

Para cumplir todo estos requisitos creé un protocolo (interfaz) que deberá ser implementado por cada uno de los test que realicemos para cumplir estos requisitos.

Aquí vemos un ejemplo del diagrama UML con tres test heredando del protocolo.

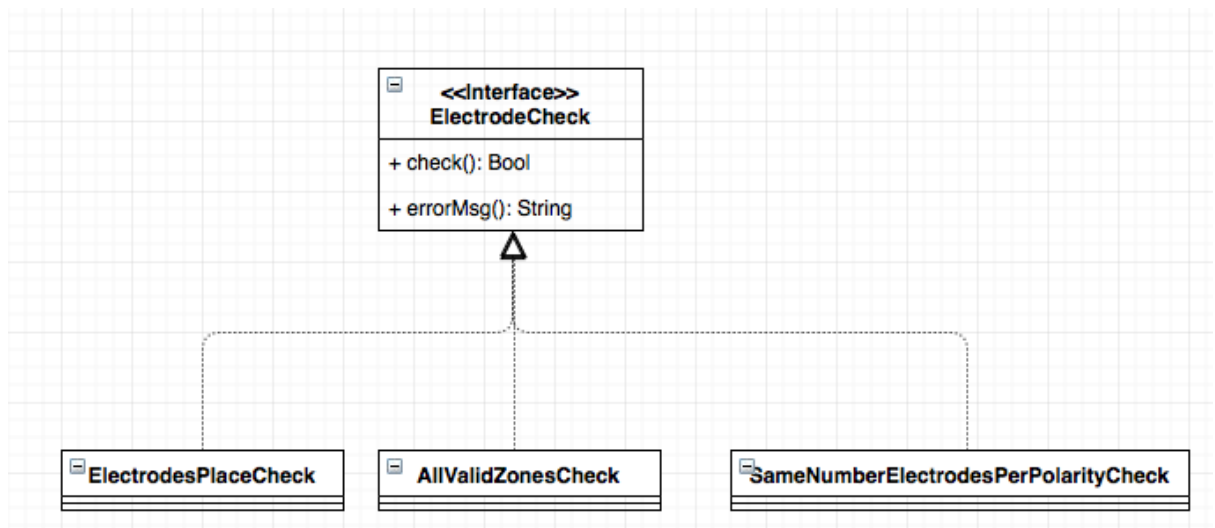


Ilustración 2.21

La función 'check' devolverá un valor booleano indicando si cumple los requisitos del test. La función 'errorMsg' devuelve el mensaje de error asociado a un test no superado. Estos son los test que se crearon:

- **ElectrodesPlacedCheck**: Comprueba que se han colocado electrodos en el modelo.
- **AllValidZonesCheck**: Comprueba que los electrodos se han colocado en la zona correcta.
- **SameNumberOfElectrodesPerPolarityCheck**: Comprueba que el número de electrodos positivos sea igual que el número de electrodos negativos.
- **TwoElectrodesPlacedCheck**: Comprueba que la técnica sea monopolar o bipolar y que se hayan colocado dos electrodos.
- **FourElectrodesInTetrapolarCheck**: Comprueba que la técnica sea tetrapolar y que se hayan colocado cuatro electrodos.
- **SameElectrodeTypeCheck**: Comprueba que la técnica es distinta de monopolar y que se hayan colocado los electrodos del mismo tamaño.

- MonopolarElectrodeSizeCheck: Comprueba que la técnica sea monopolar y que el tamaño del electrodo positivo sea mayor que el negativo.
- SegmentalBackZoneCheck: Comprueba que el método sea segmentario y que el electrodo positivo esté posicionado en la zona de la columna y el negativo en la zona a tratar.
- OneCircuitCheck: Comprueba que la técnica sea monopolar o bipolar y se haya formado un circuito.
- TwoCircuitsInTetrapolarCheck: Comprueba que la técnica sea tetrapolar y se hayan formado dos circuitos.
- CrossCircuitsInTetrapolarCheck: Comprueba que la técnica sea tetrapolar y los circuitos se crucen.
- ElectrodesCentralityCheck: Comprueba que el valor de centralidad supere un límite.
- ElectrodesOrientationCheck: Comprueba que el valor de orientación sea acorde a la técnica escogida (longitudinal o transversal).
- ElectrodesDistributionCheck: Comprueba que el valor de distribución supere un límite.

Para que el usuario pueda pasar la vista 'colocación de electrodos' y avanzar a la vista 'selección familia de corriente dispositivo' debe superar cada uno de estos test.

2.6 Sprint 6

TAREA	ESTIMACIÓN	VISTA
Mostrar mensajes de error	2	Colocación de electrodos
Añadir botones para seleccionar el modo de juego	2	Selección modo de juego
Nueva vista 'caso clínico'	2	Caso clínico
Archivo JSON para la lectura de casos clínicos	3	Caso clínico

Tabla 2.7

2.6.1 Mostrar mensajes de error

Ya tenemos los test implementados que se activarán cuando el usuario intente avanzar a la vista 'selección familia corriente de dispositivo'. Ahora vamos a mostrar el mensaje de error en caso de que el alumno se equivoque y no supere los test de comprobación de electrodos.

Para ello crearemos un objeto de la clase 'UIAlertController' con el mensaje de error del test devuelto en el método 'errorMsg'. En el siguiente ejemplo se escogió una técnica bipolar pero se colocó cuatro electrodos.

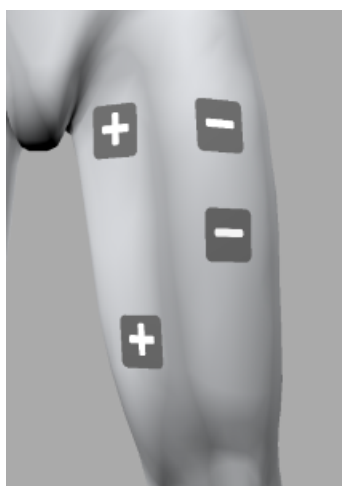


Ilustración 2.22

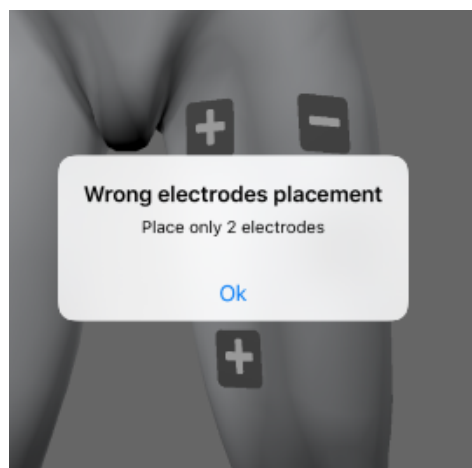


Ilustración 2.23

En este otro ejemplo se escogió la técnica bipolar y el método transversal. A pesar de que cumple la condición de la técnica bipolar, los electrodos están colocados longitudinalmente:



Ilustración 2.24

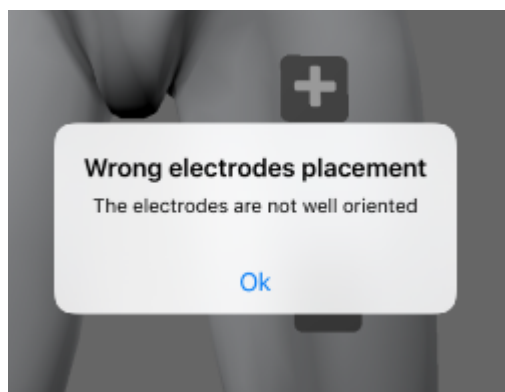


Ilustración 2.25

2.6.2 Reunión

Tras este amplio desarrollo las clientes debían constatar que tanto las técnicas como los métodos funcionaban perfectamente. Instalé la aplicación en el iPad de una de las clientes y tras una exhaustiva prueba posterior, no sólo comprobó que todo funcionaba correctamente si no que quedó muy satisfecha con el resultado. El proyecto avanzaba y el desarrollo del modo estudio había concluido. Tocaba comenzar con el objetivo principal del proyecto: el modo evaluación.

2.6.3 Añadir botones para seleccionar el modo de juego

Se cambia la primera vista ‘selección modo de juego’ para añadir dos botones (objeto UIButton) para la elección del modo de juego: aprendizaje y evaluación. Se añaden dos iconos para darle estilo a los botones además de un texto para clarificarlos.

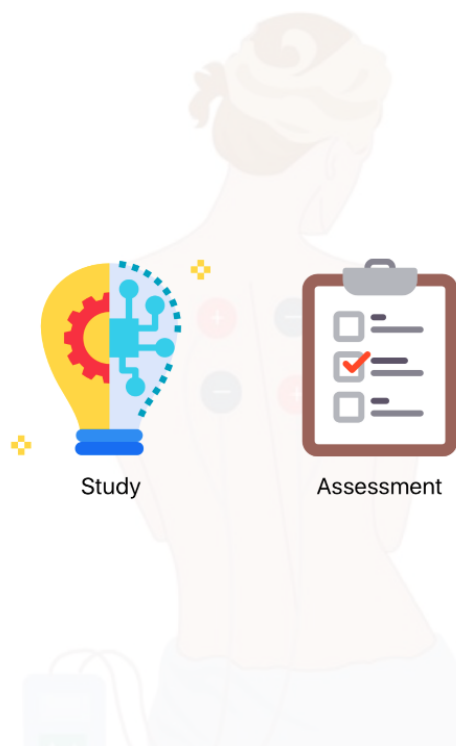


Ilustración 2.26

2.6.4 Nueva vista caso clínico

En el modo evaluación, el alumno recibirá la descripción de una dolencia de un paciente en la que se explicita toda la información necesaria para poder realizar el tratamiento correctamente. Así que creamos una nueva vista, simplemente tiene un título (objeto UILabel) y una descripción del caso clínico (objeto UILabel). Se añade una imagen de fondo para darle un poco de realismo. La idea es relacionar esta vista con un archivo JSON donde se guarde la información de cada caso clínico.



Ilustración 2.27

2.6.5 Archivo JSON para la lectura del caso clínico

Ya tenemos la vista, ahora necesitamos los datos en un archivo de almacenamiento que la aplicación pueda leer. Las clientes estuvieron trabajando en la elaboración de un caso clínico de ejemplo que me proporcionaron.

JSON es un formato de texto sencillo para el intercambio de datos. Creamos un archivo JSON con dos campos para el caso clínico: el número del caso clínico y el enunciado. El archivo quedaría de esta forma:

```
[
  {
    "caso": 1,
    "enunciado": "Paciente mujer de 41 años de edad que acude a consulta de
    fisioterapia por presentar dolor intenso en la cara interna del codo derecho
    de dos semanas de evolución. La paciente refiere que el dolor apareció a
    raíz de realizar labores de jardinería (poda de arbustos) y se intensifica
    con cualquier movimiento que implique la presión palmar (barrer, planchar y
    sobre todo retorcer algún trapo o abrir algún bote). A la exploración
    presenta fuerte dolor en la palpación del borde interno de la epitroclea,
    mantiene libres los movimientos de flexo-extensión, prono-supinación del
    codo y flexo-extensión de muñeca pero el movimiento de flexión del carpo
    contra resistencia incrementa el dolor."
  }
]
```

Ilustración 2.28

El lenguaje Swift cuenta con la funcionalidad necesaria para leer un archivo JSON sin necesidad de uso de librerías externas.

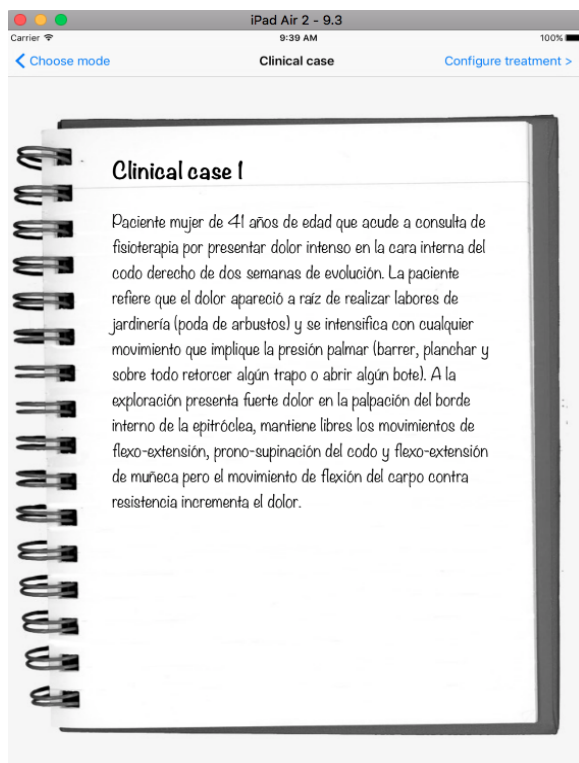


Ilustración 2.29

Creamos una clase 'ClinicCase' que encapsula un caso clínico. Cuenta con un método llamado 'readClinicCase' que a partir de la clase 'JSONSerialization' del framework 'Foundation' se encarga de deserializar los objetos JSON del modelo.

Tras enlazar los datos con la vista 'caso clínico' queda como se ve en la imagen de la izquierda.

La idea es incrementar el número de casos clínicos en un futuro, aunque antes la prioridad es conseguir evaluar este caso clínico correctamente.

2.7 Sprint 7

TAREA	ESTIMACIÓN	VISTA
Desarrollo de un controlador para los modos de juego	8	Todas

Tabla 2.8

2.7.1 Desarrollo de un controlador para los modos de juego

En el modo evaluación el alumno recibirá un caso clínico y tendrá que realizar todo el tratamiento sin ninguna ayuda por parte de la aplicación. Tras realizarlo, recibirá una puntuación.

Aunque este modo de juego difiere en algunos puntos con el modo de estudio, su parecido es mayor que su diferencia, comparten muchas vistas.

El desarrollo fácil para este modo sería duplicar aquellas vistas compartidas y así añadir la funcionalidad deseada. El problema es que duplicaríamos las vistas innecesariamente y violaríamos el principio de desarrollo DRY:

Don't repeat yourself (DRY): Cualquier parte del código nunca debe ser duplicada. La duplicación incrementa la dificultad en los cambios y la evolución posterior. Puede perjudicar la claridad del código y crear un espacio para posibles inconsistencias. Esto es muy común cuando se realiza “copy-paste” de partes de código que generan el doble de mantenimiento.

Este problema lo podemos resolver con el patrón de arquitectura ‘Modelo Vista Controlador’ (MVC) [2].

El patrón MVC separa los datos de una aplicación, la interfaz de usuario y la lógica de control (el controlador) en tres componentes distintos:

- El modelo: contiene una representación de los datos que maneja la aplicación, su lógica de negocio y sus mecanismos de persistencia.
- La vista: determina la visión y la interacción del usuario con la aplicación.
- El controlador: que actúa como intermediario entre el modelo y la vista. Transforma los datos según la necesidad de cada uno.

Veamos las vistas que comparten ambos modos de estudio y estudiemos el comportamiento que debe seguir cada uno:

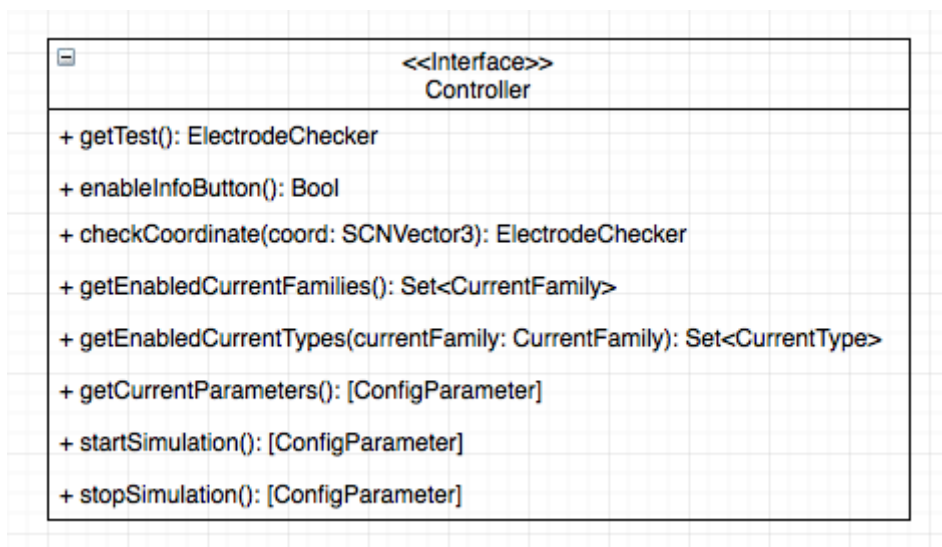
*Aunque la vista ‘selección modo de juego’ es común no hay que hacer ninguna diferenciación ya que técnicamente no se ha iniciado aún ningún modo de juego.

Vista	Funcionalidad	Modo aprendizaje	Modo evaluación
Colocación de electrodos	Mostrar mensajes de error	Sí	No
Colocación de electrodos	Mostrar información sobre el tratamiento	Sí	No
Selección familia corriente dispositivo	Deshabilitar la selección de familias erróneas	Sí	No
Selección corriente dispositivo	Deshabilitar la selección de corrientes erróneas	Sí	No
Configuración dispositivo	Poner límites en la elección de los parámetros	Sí	No
Inicio dispositivo	Poner límites al tiempo de corriente o intensidad	Sí	No
Inicio dispositivo	Mostrar la intensidad máxima	Sí	No
Inicio dispositivo	Mostrar paciente	No	Sí

Tabla 2.9

En la vista ‘selección modo de juego’ se activará el controlador del modo de juego seleccionado. En las vistas compartidas, aquella funcionalidad que difiera de un modo de juego al otro será el controlador el encargado de realizarla.

De nuevo creamos un protocolo ‘Controller’ cuyos métodos deberán ser implementadas por los controladores de ambos modos de juego.

*Ilustración 2.30*

En la tabla siguiente se describe cada uno de los métodos y la función que realizan en cada modo de juego.

Método	Modo estudio	Modo aprendizaje
getTest	Devuelve un objeto con todos los test para posteriormente ejecutarlos	Devuelve un objeto con un único test, colocar electrodos
enableInfoButton	Devuelve true para habilitar el botón con información de la vista 'colocación de electrodos'	Devuelve false para deshabilitar el botón con información de la vista 'colocación de electrodos'
checkCoordinate	Comprueba que la posición del electrodo se encuentra dentro del elipsoide de la zona a tratar	No comprueba la posición del electrodo, así que siempre devuelve true
getEnabledCurrentFamilies	Devuelve las familias de corrientes disponibles en función de la configuración realizada en la vista 'configuración del tratamiento'	Devuelve todas las familiass sin ninguna restricción
getEnabledCurrentTypes	Devuelve las corrientes disponibles en función de la	Devuelve todas las corrientes sin ninguna restricción

	configuración realizada en la vista 'configuración del tratamiento'	
getCurrentParameters	Devuelve una configuración de parámetros con límites de valores	Devuelve una configuración de parámetros sin límites de valores
startSimulation	Inicia el dispositivo	Inicia el dispositivo, guarda los parámetros configurados por el usuario e inicia el paciente
stopSimulation	Termina la simulación	Termina la simulación, guarda los parámetros configurados en esta vista y pasa a la vista 'evaluación'

Tabla 2.10

2.8 Sprint 8

TAREA	ESTIMACIÓN	VISTA
Crear niveles de sensibilidad	2	Inicio de dispositivo
Inclusión de un paciente	3	Inicio de dispositivo
Spike para la evaluación	3	

Tabla 2.11

2.8.1 Reunión

Nos reunimos las product owners y yo para mostrar los avances de la aplicación. Me comentan si hay posibilidad de incluir una tabla de mensajes o algo parecido en la vista 'inicio dispositivo' para guiar al alumno a la hora de aplicar la intensidad. La intensidad no es un valor establecido como puede ser la frecuencia de corriente, ya que depende del paciente y de su nivel de sensibilidad.

Los niveles de sensibilidad son intervalos de valores de intensidad con un mínimo y un máximo. Tenemos los siguientes tipos:

- Subliminal: Ninguna sensación.
- Liminal: Sensación leve como un hormigueo.
- Supraliminal motor: Contracciones musculares espontáneas. La sensación empieza a ser fuerte.
- Supraliminal: Sensación fuerte pero sin dolor.
- Máximo: Sensación fuerte casi insoportable.
- Dolor: Dolor, posibles quemaduras.

Aunque los valores para estos intervalos son diferentes en función del paciente, hay algo en común: la sensación. Por lo tanto, el mensaje de comunicación del paciente con el especialista será muy parecido en todos los casos.

2.8.2 Crear niveles de sensibilidad

Un nivel de sensibilidad se puede visualizar como un trapecio, cuyos puntos superiores son el valor mínimo y máximo de intensidad para ese intervalo. Las “rampas” de los trapecios se podrían definir como el nivel de resistencia del paciente para entrar o salir del nivel de sensibilidad (Ilustración 2.31).

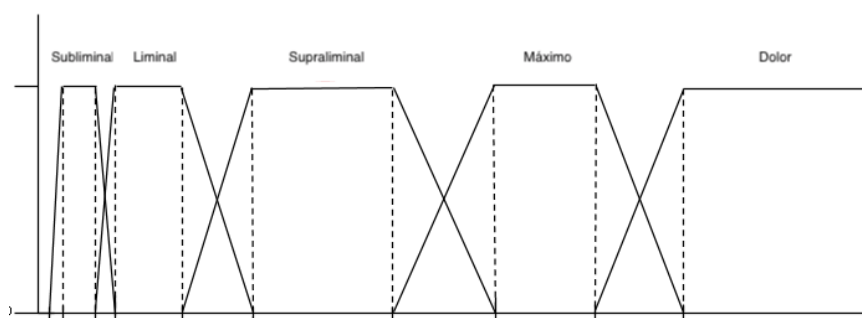


Ilustración 2.31

Tenemos diferentes valores en las rampas: más suaves o más inclinadas. Diferentes valores de separación entre el valor mínimo y el máximo (puntos superiores) de cada intervalo. Estos valores los generamos de forma aleatoria para que cambien con cada paciente y así no tener unos valores establecidos de intensidad que el alumno podría memorizar. De esta forma hacemos este proceso lo más realista posible.

2.8.3 Inclusión de un paciente

Esta aplicación es un serious game, así que al fin y al cabo, no deja de ser un juego. Hay que intentar hacerlo entretenido, interactivo y dinámico. Pensé en incluir a un paciente con imágenes y bocadillos para comunicarse con el alumno de la forma más realista posible.

Añadimos dos imágenes (UIImageView). Una imagen para el paciente que cambiará con la intensidad y otra para el bocadillo. Dentro del bocadillo tenemos una etiqueta (UILabel) que mostrará el mensaje.

Creamos una clase ‘GamePatient’ que encapsula al paciente. Esta clase será la encargada de crear los niveles de sensibilidad aleatoriamente y de proporcionar los mensajes en función de la intensidad. Es importante resaltar que los niveles de

sensibilidad dependen también del tipo de corriente (en el Anexo 3 está la tabla con los niveles de sensibilidad relacionados con las corrientes y los mensajes).



Ilustración 2.32

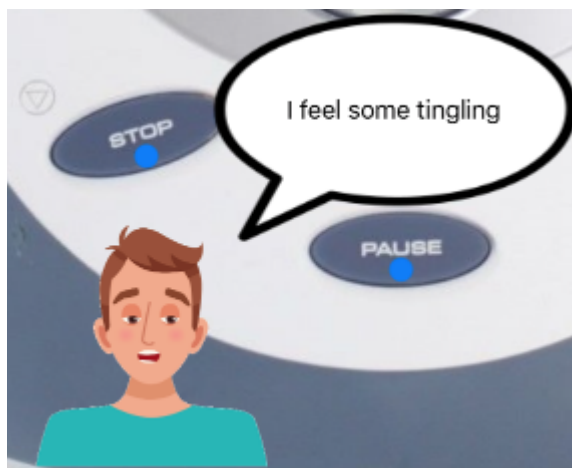


Ilustración 2.33



Ilustración 2.34



Ilustración 2.35



Ilustración 2.36

2.8.4 Spike para la evaluación

Realizar una evaluación de todo el proceso, desde la elección del objetivo terapéutico hasta los valores de frecuencia, intensidad o tiempo de corriente, pasando por la colocación de los electrodos, es extremadamente difícil.

Hay variables como la elección de la técnica, que dado un caso clínico sólo se pueda escoger una, por ejemplo, bipolar. Pero para la frecuencia, por poner otro ejemplo, puede haber un intervalo de valores que sea 100% correcto, pero otro rango de valores que tal vez no sea el más adecuado pero que tampoco es erróneo y sea 50% correcto.

Tenemos una gran cantidad de variables y diferentes valores de puntuación para cada variable, por ello que la evaluación no sea nada fácil. Hicimos una división de las variables en dos grupos:

- No difusas: Variables cuyo valor es correcto o incorrecto, no hay término medio. Por ejemplo: la elección de zona, el método o la técnica usada, el tipo de electrodos usados, etc.
- Difusas: Variables cuyo valor no se puede definir como correcto o incorrecto exclusivamente. Por ejemplo: frecuencia, tiempo de tratamiento, tiempo de impulso, intensidad...

Para la evaluación de las variables difusas incorporamos al equipo de desarrollo a Carmen Martínez Cruz, experta en lógica difusa. Ella se encargará de coordinar las tareas para la evaluación difusa.

Para la parte de la evaluación no difusa podemos reusar los test implementados en el modo estudio que nos indicarán si variables como la técnica, método o demás están bien usados por parte del alumno o no.

2.9 Sprint 9

TAREA	ESTIMACIÓN	VISTA
Añadir solución al archivo JSON	2	Evaluación
Creación del archivo FuzzyLite	5	Evaluación

Tabla 2.12

2.9.1 Añadir solución al archivo JSON

Las clientes me pasaron un pdf con la solución del tratamiento del primer caso clínico que incluimos en el archivo JSON. En la solución aparecen los valores correctos para cada una de las variables: técnica, método, zona, tipo de electrodos, frecuencia, tiempo de tratamiento, nivel de sensibilidad...

Se decide incluir en el archivo JSON la solución para aquellas variables no difusas, las que vamos a evaluar con los test, ya que las difusas la vamos a evaluar de una forma diferente propuesta por Carmen. Así que tras incluir la solución, el archivo JSON queda así:

```
[
  {
    "caso": 1,
    "enunciado": "Paciente mujer de 41 años de edad que acude a consulta de fisioterapia por presentar dolor intenso en la cara interna del codo derecho de dos semanas de evolución. La paciente refiere que el dolor apareció a raíz de realizar labores de jardinería (poda de arbustos) y se intensifica con cualquier movimiento que implique la presión palmar (barrer, planchar y sobre todo retorcer algún trapo o abrir algún bote). A la exploración presenta fuerte dolor en la palpación del borde interno de la epitroclea, mantiene libres los movimientos de flexo-extensión, prono-supinación del codo y flexo-extensión de muñeca pero el movimiento de flexión del carpo contra resistencia incrementa el dolor.",
    "zona": "codoFD",
    "metodo": ["transversal", "longitudinal"],
    "técnica": ["monopolar", "bipolar"],
    "electrodos": ["60x80", "50x50", "32"],
    "configuracion_corriente": [{"corriente": "tens_asimetricabifasica", "cccv": "cc", "polaridad": "n", "modulacion": "s"}, {"corriente": "interferencial_bipolar", "cccv": "cc", "polaridad": "n", "modulacion": "s"}]
  }
]
```

Ilustración 2.37

Además del número del caso clínico y del enunciado, se incluye la solución no difusa:

- La zona a tratar.
- El método, puede ser válido más de uno.
- La técnica, puede ser válido más de una.
- El tipo de electrodos, puede ser válido más de uno.
- La configuración de la corriente, puede ser válida más de una. Dentro de la configuración:
 - Corriente
 - CC/CV
 - Polaridad
 - Modulación

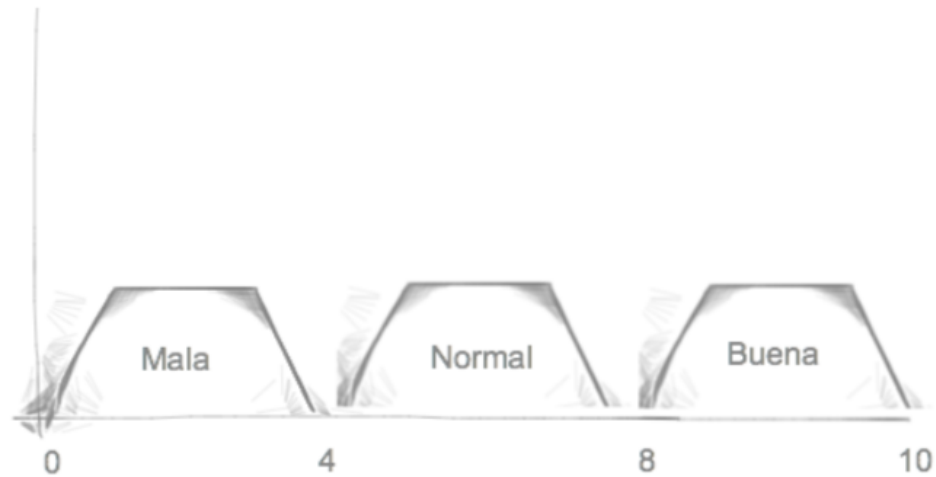
2.9.2 Creación del archivo Fuzzylite

En una reunión con Carmen Martínez, experta en lógica difusa, expone cómo vamos a realizar la evaluación de las variables difusas. Usaremos una librería externa llamada FuzzyLite.

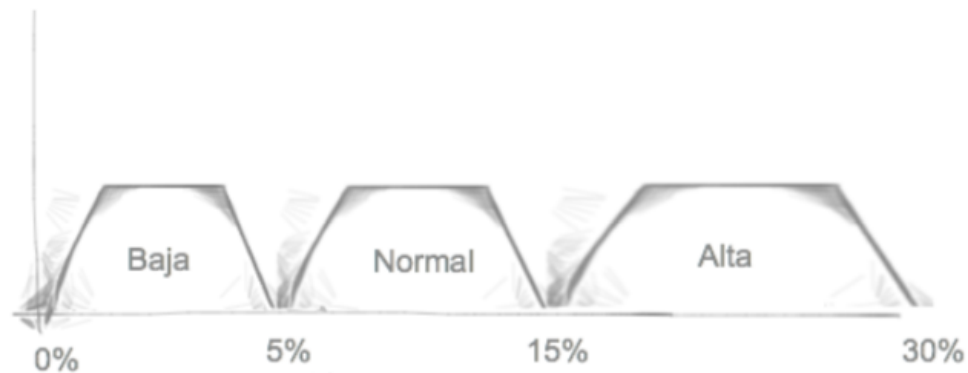
FuzzyLite es una librería de control de lógica difusa gratuita y de código abierto programada en C++ para múltiples plataformas.

Veamos la definición que hay en Wikipedia de la lógica difusa: *“La lógica difusa (también llamada lógica borrosa) se basa en lo relativo de lo observado como posición diferencial. Este tipo de lógica toma dos valores aleatorios, pero contextualizados y referidos entre sí”*. Todo en esta vida se ve más fácil con un ejemplo:

Vamos a un restaurante y pensamos en la propina que vamos a dejar: baja (0% - 5% de la cuenta), normal (5% - 15%) o alta (15% - 30%). Tenemos dos variables que condicionan la propina: calidad de la comida, y calidad del servicio. Para cada variable tres valores: mala (0 - 4), normal (4 - 8) y buena (8 - 10).

*Ilustración 2.38*

Variable: calidad de la comida y calidad del servicio

*Ilustración 2.39*

Variable: propina

Puntuamos cada variable del 0 al 10. Este valor cuantitativo se convierte en cualitativo.

Definimos varias reglas, por ejemplo:

- Si la comida es buena y la calidad del servicio mala -> propina normal.
- Si la comida es buena y la calidad del servicio buena -> propina alta.
- Si la comida es mala y la calidad del servicio mala -> propina baja.

Con los valores de la puntuación obtenemos un tanto por ciento de la propina que vamos a dejar.

Así funciona la lógica difusa y esto mismo queremos extrapolarlo a nuestro problema de la puntuación en el modo evaluación. Tenemos muchas más variables y muchos más valores pero el funcionamiento es similar. Esto es lo que nos permite la librería FuzzyLite.

Añadimos los archivos de cabecera .h y los archivos cuerpo .cpp de la librería al proyecto para comenzar a hacer uso de ella. Carmen estuvo trabajando en la elaboración de un archivo en el que se describe cada uno de los objetos geométricos que determinan el valor de la variable. Cada objeto geométrico (trapezio, triángulo, etc.) se define con los valores asociados a una variable, y si es necesario, a un tipo de corriente ya que cada valor de una variable del dispositivo cambia con el tipo de corriente. Veamos una parte del archivo:

```
InputVariable: ORIEN
enabled: false
range: 0 1
lock-range: false
term: Transversal Trapezoid 0 0 0.2 0.35
term: Longitudinal Trapezoid 0.65 0.8 1 1
```

```
InputVariable: CENTR
enabled: false
range: 0 1
lock-range: false
term: Right Triangle 0 0 0.75
```

```
InputVariable: DISTR
enabled: false
range: 0 1
lock-range: false
term: Right Triangle 0.25 0.6 0.95
```

Ilustración 2.40

Aquí se definen las variables que definen la calidad en la colocación de los electrodos: orientación, centralidad y distribución. Cada variable lleva asociado una geometría, cada geometría se define con unos puntos. Por ejemplo, el trapecio “Transversal” de la variable orientación se define con estos valores: 0 - 0 - 0.2 - 0.35.

En el proceso de la evaluación difusa que realizaremos posteriormente le proporcionaremos el valor de orientación que el usuario ha conseguido, si ese valor se encuentra entre 0 y 0.2 significa que ha colocado los electrodos de manera transversal 100% correcta. Si el valor devuelto se encuentra en el intervalo [0.2 - 0.35] significa que la colocación es transversal pero no 100% correcta.

Esto es extensible a las variables de centralidad y distribución aunque el objeto geométrico es un triángulo. Ahora para alcanzar el valor 100% habrá que acercarse a la cúspide, 0 para el caso de la centralidad y 0.6 para la distribución.

La idea es almacenar los valores de las variables que el usuario va consiguiendo a lo largo del proceso y compararlos con los definidos en este archivo. Esto lo hace automáticamente la librería y recibiremos una nota por cada variable.

2.10 Sprint 10

TAREA	ESTIMACIÓN	VISTA
Desarrollo evaluación difusa	8	Evaluación

Tabla 2.13

2.10.1 Desarrollo evaluación difusa

Ya tenemos el archivo FuzzyLite “FisioDesc_v7.fl” listo para la evaluación difusa. Ahora hay que agrupar todas las variables difusas que configura el usuario. Estas son:

- Orientación: valor de la orientación de los electrodos en la vista ‘colocación de electrodos’.
- Centralidad: valor de la centralidad de los electrodos en la vista ‘colocación de electrodos’.
- Distribución: valor de la distribución de los electrodos en la vista ‘colocación de electrodos’.
- Frecuencia: valor de la frecuencia establecido en la vista ‘configuración dispositivo’.
- Tiempo de impulso: valor del tiempo de impulso establecido en la vista ‘configuración dispositivo’.
- Tiempo de tratamiento: valor del tiempo de tratamiento establecido en la vista ‘inicio dispositivo’.
- Intensidad: valor de la intensidad establecido en la vista ‘inicio dispositivo’.

Creamos una clase estática ‘GameResults’ que encapsula todos las variables que serán evaluadas. El controlador será el encargado de ir almacenando los valores en esta clase.

Creamos otra clase llamada 'GameEvaluation' con el método "fuzzyEvaluation" que realizará la evaluación difusa siguiendo estos pasos:

- Lee el archivo "FisioDesc_v7.fl".
- Se crea un sufijo en función de la corriente seleccionada. Este sufijo nos sirve para buscar la geometría en el archivo. Ya que cada geometría depende de la variable y de la corriente.
- Para cada una de las variables a evaluar se crea una "variable de entrada".
- Para cada una de las variables de entrada se introduce el valor que tenemos en la clase estática 'GameResults'.
- Se procesan las variables. Para cada una de las variables de entrada obtenemos la evaluación que guardaremos en los atributos de esta misma clase.

2.11 Sprint 11

TAREA	ESTIMACIÓN	VISTA
Desarrollo evaluación no difusa	8	Evaluación
Nueva vista 'evaluación'	2	Evaluación

Tabla 2.14

2.11.1 Desarrollo evaluación no difusa

Las variables no difusas que hay que evaluar son las siguientes:

- Zona: zona tratada, donde se colocan los electrodos.
- Técnica: relacionado con el número de electrodos y el tamaño.
- Método: relacionado con la forma de colocar los electrodos.
- Tipos de electrodos: electrodos con diferentes tamaños.
- Polaridad: mismo número de polaridad de electrodos.
- Corriente: tipo de familia y corriente seleccionada.

Podemos reutilizar los test implementados para la comprobación de errores en la vista 'colocación de electrodos' del modo estudio. Estos test devuelven un booleano que nos indican si se cumple o no la condición. Aplicando los test correctos para cada variable podemos obtener una evaluación. Estas variables sólo pueden tener dos soluciones: correcto o incorrecto. No hay término medio como en las variables difusas.

Veamos en pseudocódigo como se evaluaría la variable 'técnica':

PARA cada una de las técnicas

HACER:

caso 'Bipolar':

```
esBipolar <- TwoElectrodesPlacedCheck
              AND OneCircuitCheck
              AND SameElectrodeTypeCheck
```

caso 'Monopolar':

```
esMonopolar <- TwoElectrodesPlacedCheck
                AND OneCircuitCheck
                AND MonopolarElectrodesSizeCheck
```

caso 'Tetrapolar':

```
esTetrapolar <- FourElectrodesInTetrapolarCheck
                AND TwoCircuitsInTetrapolarCheck
                AND CrossCircuitsInTetrapolarCheck
```

FIN HACER

```
tecnica <- esBipolar OR esMonopolar OR esTetrapolar
```

```
tecnicaSolucion <- leerTecnicaArchivoJSON()
```

DEVOLVER (tecnicaSolucion IGUAL QUE técnica)

Esta solución es extensible a las demás. Los pasos son los mismos:

- Aplicar test para averiguar la solución del alumno
- Leer la solución del archivo JSON
- Comparar el resultado del alumno con la solución

2.11.2 Nueva vista evaluación

Añadimos una vista final en la que el usuario recibirá una evaluación del proceso de tratamiento realizado. Cada apartado es una variable que se puntúa con un icono para el caso de las variables no difusas y con un icono y un porcentaje para el caso de las variables difusas. La vista está subdividida en tres secciones:

- Colocación de electrodos:
 - Elección de zona
 - Método
 - Técnica
 - Tipo de electrodos
 - Polaridad
 - Orientación
 - Centralidad
 - Distribución
- Configuración de corriente:
 - Corriente
 - Tiempo de impulso
 - Frecuencia
 - Polaridad
 - Modulación
 - Intensidad
 - Tiempo de tratamiento

Treatment assessment		
ELECTRODE PLACEMENT		
Choice of zone		✓
Method		✓
Technique		✓
Electrodes type		✓
Electrodes polarity		✓
Zone orientation	100 %	✓
Zone distribution	2 %	✗
Zone centrality	26 %	✗
CURRENT CONFIGURATION		
Current		✓
Impulse time	100 %	✓
Frequency	100 %	✓
Polarity		✓
Modulation		✓
Intensity	100 %	✓
Time of treatment	100 %	✓
OVERALL SCORE		
Final score	94 %	✓

Ilustración 2.41

- Puntuación final

2.12 Sprint 12

TAREA	ESTIMACIÓN	VISTA
Desarrollo para el cálculo de la nota final	2	Evaluación
Añadir nuevos casos clínicos	2	Caso clínico

Tabla 2.15

2.12.1 Desarrollo para el cálculo de la nota final

Para el cálculo de la nota final no ha habido un consenso suficiente para llegar a una solución clara hasta el momento de la redacción de este documento. Había que terminarlo de alguna forma, así que opté por estudiar yo mismo una forma para evaluar la gran cantidad de variables del proceso. Lo que sí tenemos claro es que el alumno debe suspender cuando:

- La colocación de los electrodos no se realiza en la zona correcta.
- El usuario quema al paciente debido a alta intensidad.

Hasta ahora tenemos una puntuación por cada variable difusa o no difusa. Con esta información realicé una evaluación final asociada a porcentajes. Cada variable supone un porcentaje de importancia en la nota final. Así variables como la técnica o el método tendrán más peso que variables con menos importancia como los valores de calidad en la colocación de electrodos que son un tanto restrictivos. Los porcentajes son:

- Elección de la zona: 5%
- Método: 12.5%
- Técnica: 12.5%
- Tipo de electrodos: 5%
- Polaridad de electrodos: 5%
- Orientación: 3.33...%
- Centralidad: 3.33...%
- Distribución: 3.33...%
- Corriente: 10%

- Tiempo de impulso: 5%
- Frecuencia: 5%
- Polaridad: 5%
- Modulación: 5%
- Intensidad: 15%
- Tiempo de tratamiento: 5%

Una vez tengamos la nota de cada variable realizada en la evaluación difusa y no difusa, para calcular la nota final habría que multiplicar a cada puntuación su porcentaje. La suma de cada valor es la nota final. Si la puntuación de alguna de las dos variables críticas está suspensa, el alumno debe suspender. Es importante resaltar que no todos los casos clínicos cuentan con estas variables de puntuación. En el caso de tener menos variables habría que normalizar la nota final sobre diez.

2.12.2 Añadir nuevos casos clínicos

Las clientes elaboraron un documento con trece casos clínicos más con problemas completamente diferentes. Realizaron un trabajo fundamental para la aplicación y es que de poco servía tener un solo caso clínico.

Además, esto fue una buena forma de testear la aplicación y comprobar que el desarrollo realizado era robusto y de buena calidad. Para ello simplemente había que incluir cada caso clínico en el archivo JSON, si la evaluación para cada caso clínico era la correcta significaba que nuestra aplicación estaba lista para empezar a ser usada en el grado de fisioterapia.

2.13 Pruebas finales

Tras el desarrollo, instalé la aplicación en el dispositivo iPad de una cliente. Iba a realizar un testeo profundo para probar que no hubiera fallos o incongruencias con la evaluación. Además proporcionaría sugerencias y/o propuestas futuras.

Aquí la lista de bugs detectados y propuestas:

Al seleccionar la familia galvánica interrumpida la aplicación se bloqueaba	Solucionado
Colocar electrodos en más de una zona	Sin hacer, implica nuevo desarrollo
Poder cambiar el valor de CC/CV	Solucionado
La opción de modulación no debe aparecer en todas las corrientes	Solucionado
Cambio de variables excluyentes (las que hay que aprobar obligatoriamente)	Solucionado
Cambio en la manera de calcular la nota final	Solucionado
Cambio unidades: 'ms' por 'µs'	Solucionado
Cambio en los mensajes del paciente	Sin hacer, falta de consenso
Añadir botón repetir caso clínico	Sin hacer, habría que hablarlo
Cambio unidades: 'mm2' a 'cm2'	Solucionado

Tabla 2.16

En la lista había dos tareas que implicaban un amplio desarrollo y que no concuerdan con los requisitos pedidos en sprint anteriores. Por ejemplo, colocar los electrodos en varias zonas simultáneamente. Esta tarea se realizó en los primeros sprints y no se especificó nada con respecto a esto.

En cuanto al cambio de los mensajes, no es tan complicado. El problema es que en el momento de la definición de esta tarea, los mensajes eran otros.

Los bugs y cambios menores fueron solucionados.

3 EXPERIMENTACIÓN, RESULTADOS Y DISCUSIÓN

3.1 Experimentaciones y pruebas

La aplicación estaba operativa para su uso con alumnos. Había que valorar el uso de la aplicación como herramienta de aprendizaje autónomo y de evaluación.

La profesora del grado de fisioterapia de la Universidad de Jaén María Catalina Osuna redactó un cuestionario de valoración (ver anexo “Cuestionario de opinión”) para que Ángeles Díaz, profesora también de fisioterapia de la Universidad de Jaén proporcionara a sus alumnos en las clases de electroterapia.

Contaban con dos dispositivos iPad, para la recogida de datos y estadísticas hicieron uso de una aplicación móvil llamada “Socrative”.

3.2 Resultados y discusión

En el anexo “Resultados cuestionario de opinión” podemos ver el resultado del cuestionario. Lo realizaron 42 alumnos del grado de fisioterapia. La nota media del cuestionario de opinión es 7.1 puntos sobre 10.

Los resultados son muy gratificantes y esperanzadores. A pesar de que aún hay que pulir ciertos temas, creo que el trabajo realizado por parte del equipo ha sido excepcional.

4 MANUAL DE USUARIO

4.1 Storyboard

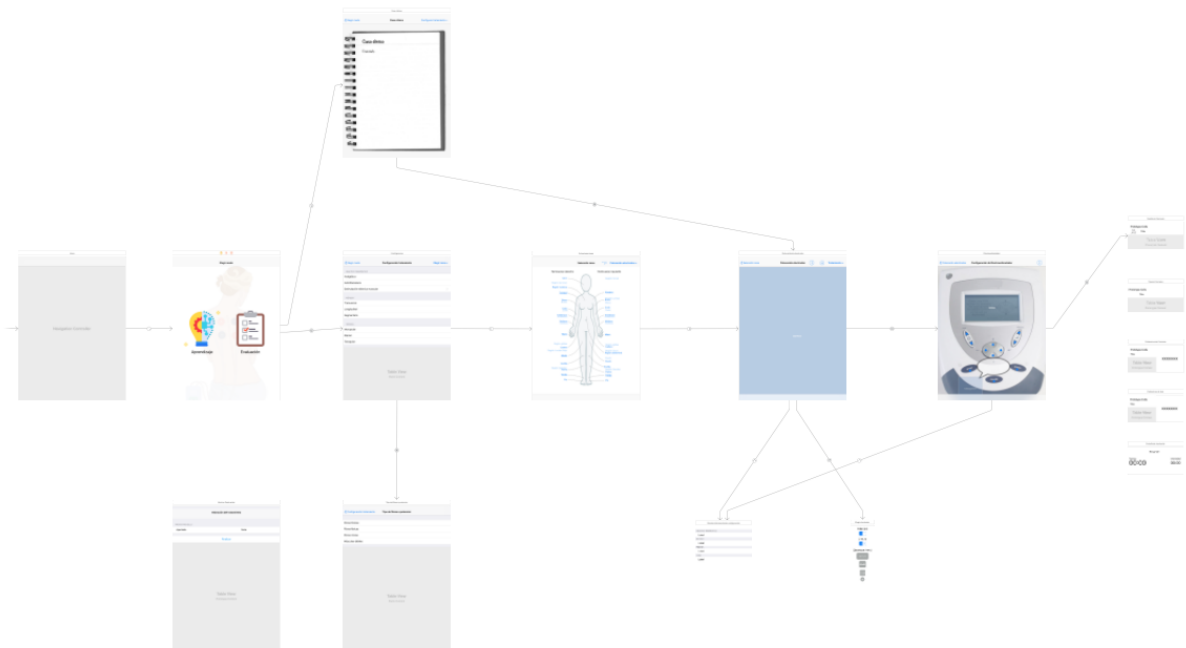


Ilustración 4.1

Estas son las vistas de la aplicación. Unas pertenecen exclusivamente a uno modo de juego y otras vistas son comunes:

- Vista selección modo de juego: Común.
- Vista configuración de tratamiento: Modo aprendizaje.
- Vista caso clínico: Modo evaluación.
- Vista configuración del tratamiento: Modo aprendizaje.
- Vista selección de zona: Modo aprendizaje.
- Vista colocación de electrodos: Común.
- Vistas relacionadas con el dispositivo: Común.
- Vista evaluación: Modo evaluación.

4.2 Vista selección modo de juego

En esta vista se selecciona el modo de juego, dos opciones:

- Modo aprendizaje: El usuario podrá seleccionar una configuración del tratamiento y la aplicación le guiará a través del proceso para poder realizarlo correctamente. Para ello habrá opciones limitadas y mensajes de error impidiendo que el alumno avance hasta que realice cada paso correctamente.
- Modo evaluación: El usuario recibirá un caso clínico y tendrá total libertad para poder tratar al paciente según su criterio. Tras finalizar recibirá una evaluación con un desglose de la puntuación dividida en varios apartados.

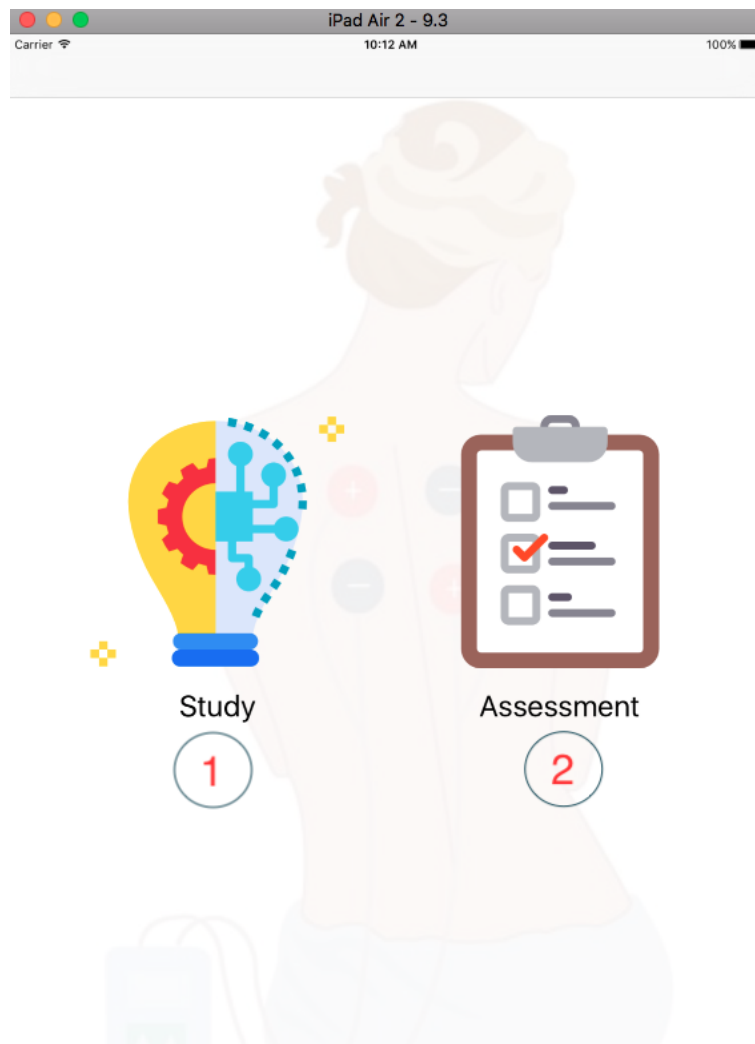


Ilustración 4.2

1 -> Avanza a la vista 'configuración del tratamiento'.

2 -> Avanza a la vista 'caso clínico'.

4.3 Modo aprendizaje

4.3.1 Vista configuración del tratamiento

En esta vista el usuario podrá elegir la configuración del tratamiento con la que desea practicar, esto servirá para parametrizar el tratamiento y establecer las restricciones para que el usuario realice el proceso correctamente.

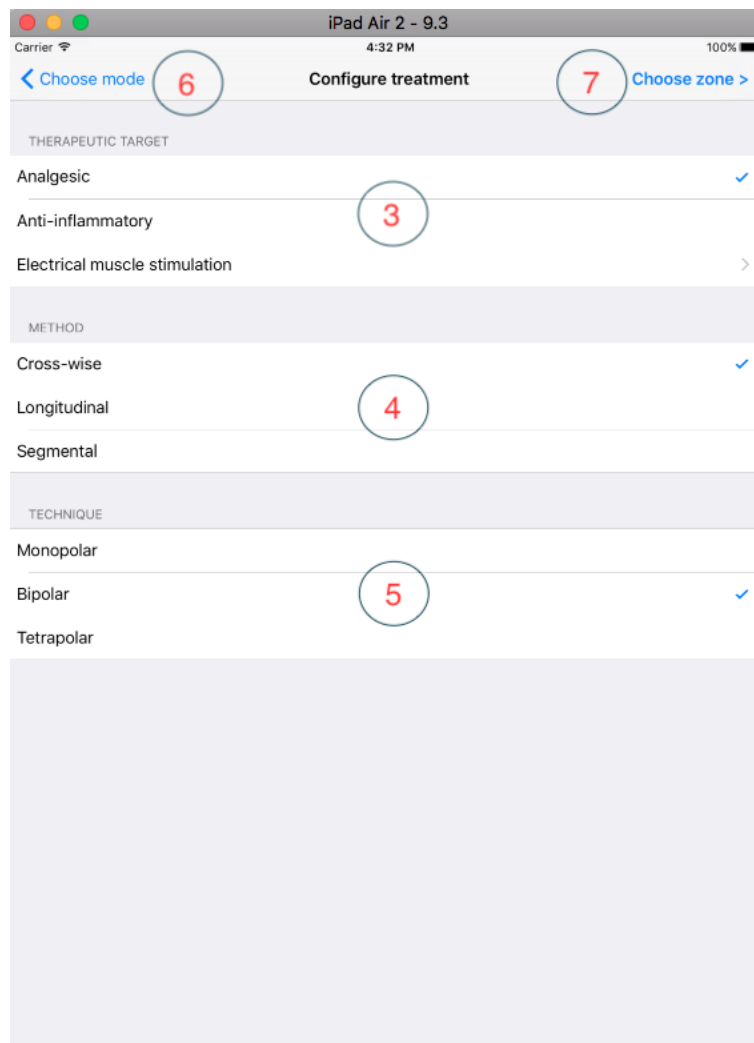


Ilustración 4.3

3 -> Permite establecer el objetivo terapéutico.

4 -> Permite establecer el método.

5 -> Permite establecer la técnica.

6 -> Retrocede a la vista 'selección modo de juego'.

4.3.2 Vista selección de zona

En esta vista el usuario seleccionará la zona a tratar con la que quiere practicar.

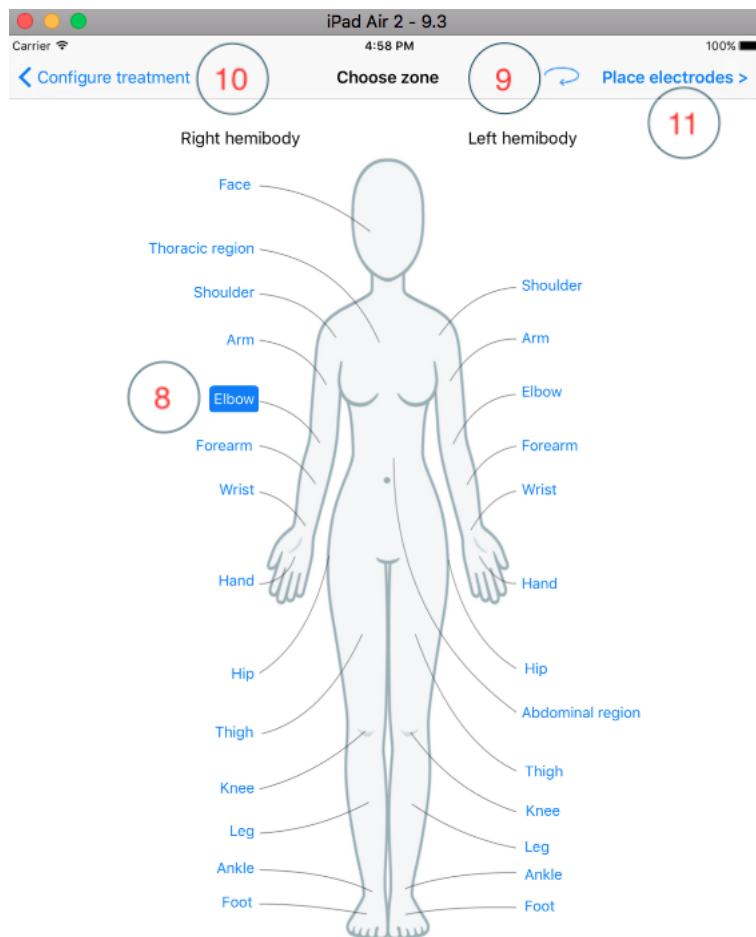


Ilustración 4.4

8 -> Selecciona la zona.

9 -> Invierte el lado del cuerpo (delantero / trasero).

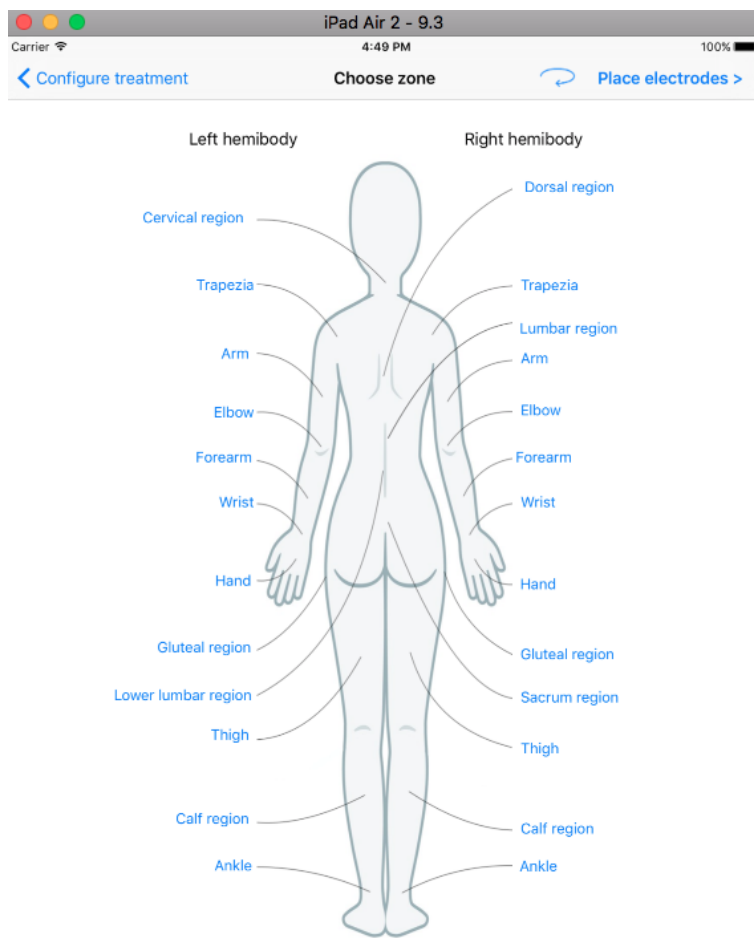


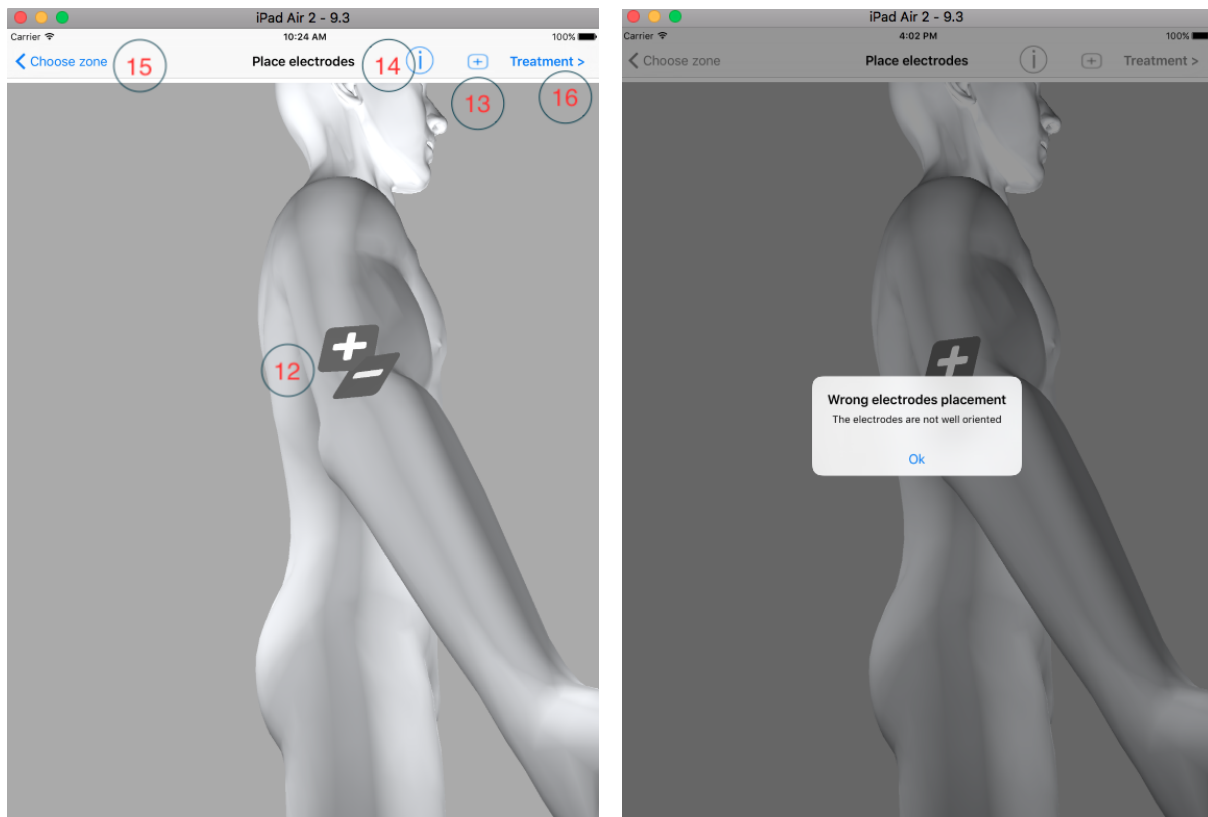
Ilustración 4.5

10 -> Retrocede a la vista 'configuración tratamiento'.

11 -> Avanza a la vista 'colocación de electrodos'.

4.3.3 Vista colocación de electrodos

En esta vista el usuario podrá colocar los electrodos en la posición/orientación/distribución que considere oportuno. Si existe algún error se mostrarán mensajes de ayuda para que alumno coloque los electrodos correctamente.



12 -> Coloca/mueve electrodos.

13 -> Abre la ventana de la configuración de los electrodos. Se puede cambiar la polaridad del electrodo, el número del circuito y el tamaño del electrodo.

14 -> Abre una ventana con información sobre los parámetros de tratamiento escogidos en la vista de configuración del tratamiento.

15 -> Retrocede a la vista de 'selección de zona'.

16 -> Avanza a la vista 'selección familia corriente dispositivo'.

4.3.4 Selección familia corriente dispositivo

En esta vista el usuario podrá seleccionar el tipo de familia. Las familias seleccionables dependen de la configuración realizada en la vista 'configuración del tratamiento'.



Ilustración 4.6

17 -> Permite subir o bajar la selección.

18 -> Retrocede a la vista 'colocación de electrodos'.

19 -> Avanza a la vista 'selección corriente dispositivo'.

4.3.5 Vista selección corriente dispositivo

Hay algunos tipos de familias de corrientes que tienen a su vez tipos de corrientes. En esta vista el usuario podrá seleccionarlas en el caso de haberlas.

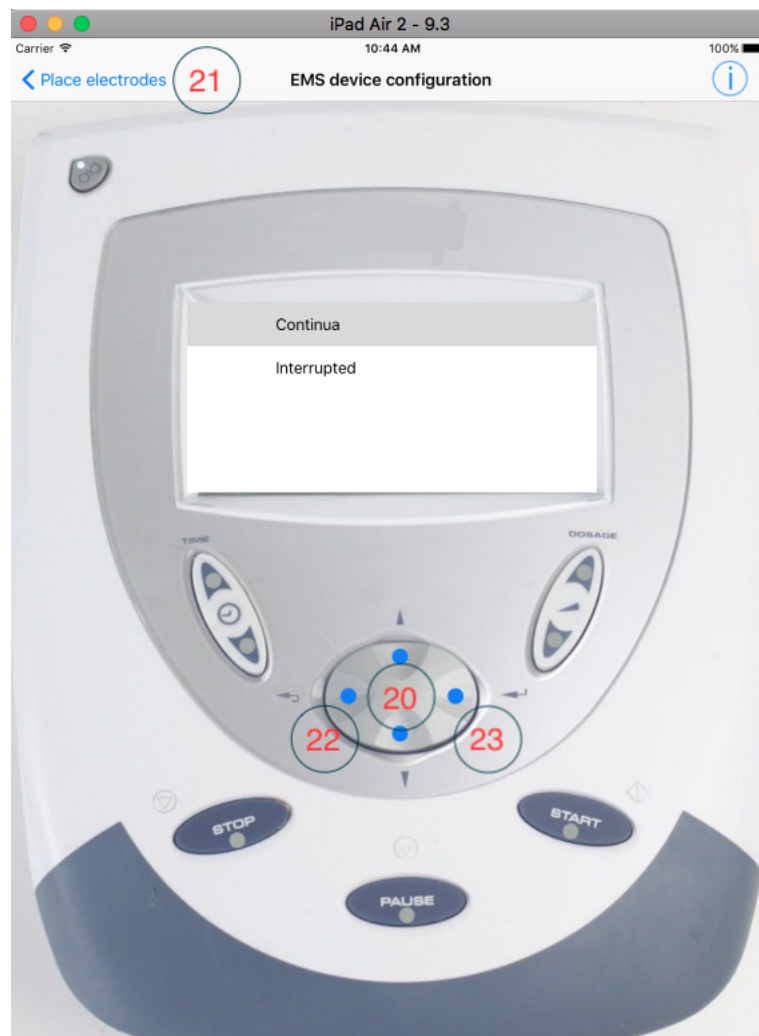


Ilustración 4.7

20 -> Permite subir o bajar la selección.

21 -> Retrocede a la vista 'colocación de electrodos'.

22 -> Retrocede a la vista 'selección familia corriente dispositivo'.

23 -> Avanza a la vista 'configuración dispositivo'.

4.3.6 Vista configuración dispositivo

En esta vista el usuario podrá configurar los parámetros del dispositivo pero con ciertas restricciones en los valores.



Ilustración 4.8

24 -> Permite subir o bajar la selección.

25 -> Permite cambiar el valor de la selección.

26 -> Retrocede a la vista 'colocación de electrodos.'

27 -> Retrocede a la vista 'selección corriente dispositivo'.

28 -> Avanza a la vista 'inicio dispositivo'.

4.3.7 Vista inicio dispositivo

En esta vista el usuario iniciará el dispositivo pudiendo cambiar los valores de intensidad y tiempo de tratamiento antes o después del inicio del dispositivo.



Ilustración 4.9

29 -> Permite cambiar el tiempo de tratamiento.

30 -> Permite cambiar el valor de la intensidad de corriente.

31 -> Inicia el dispositivo. Permite además pausar el tiempo y parar el dispositivo.

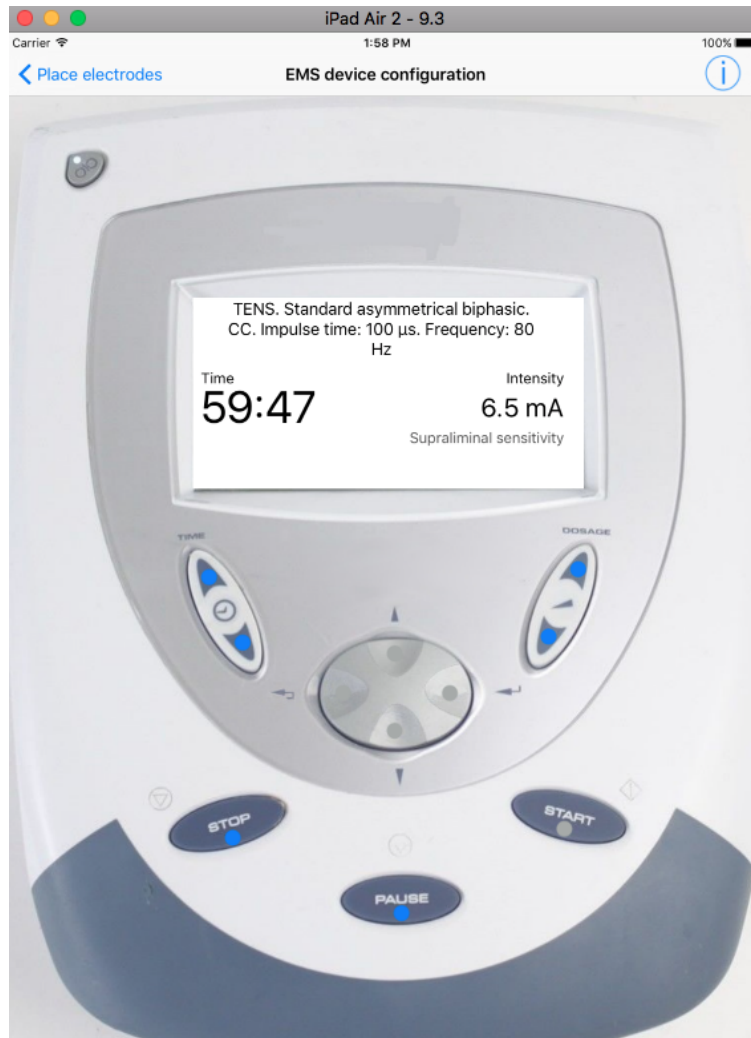


Ilustración 4.10

32 -> Retrocede a la vista 'colocación de electrodos'.

33 -> Retrocede a la vista 'configuración dispositivo'.

4.4 Modo evaluación

4.4.1 Vista caso clínico

En esta vista el usuario recibirá un caso clínico en el que se describe las molestias de un paciente en una determinada zona del cuerpo así como más información para poder llevar a cabo el tratamiento correctamente.

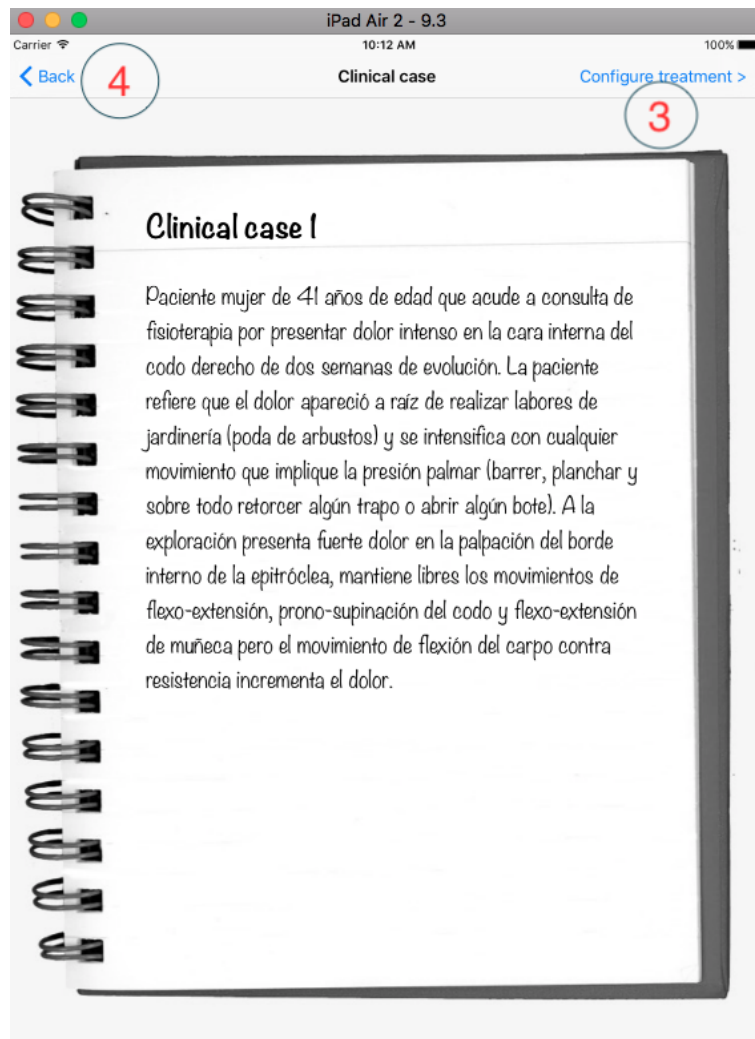


Ilustración 4.11

3 -> Avanza a la vista 'colocación de electrodos'.

4 -> Retrocede a la vista de 'selección modo de juego'.

4.4.2 Vista colocación de electrodos

En esta vista el usuario podrá colocar los electrodos en la posición/orientación/distribución que considere oportuno sin restricciones de ninguna clase.

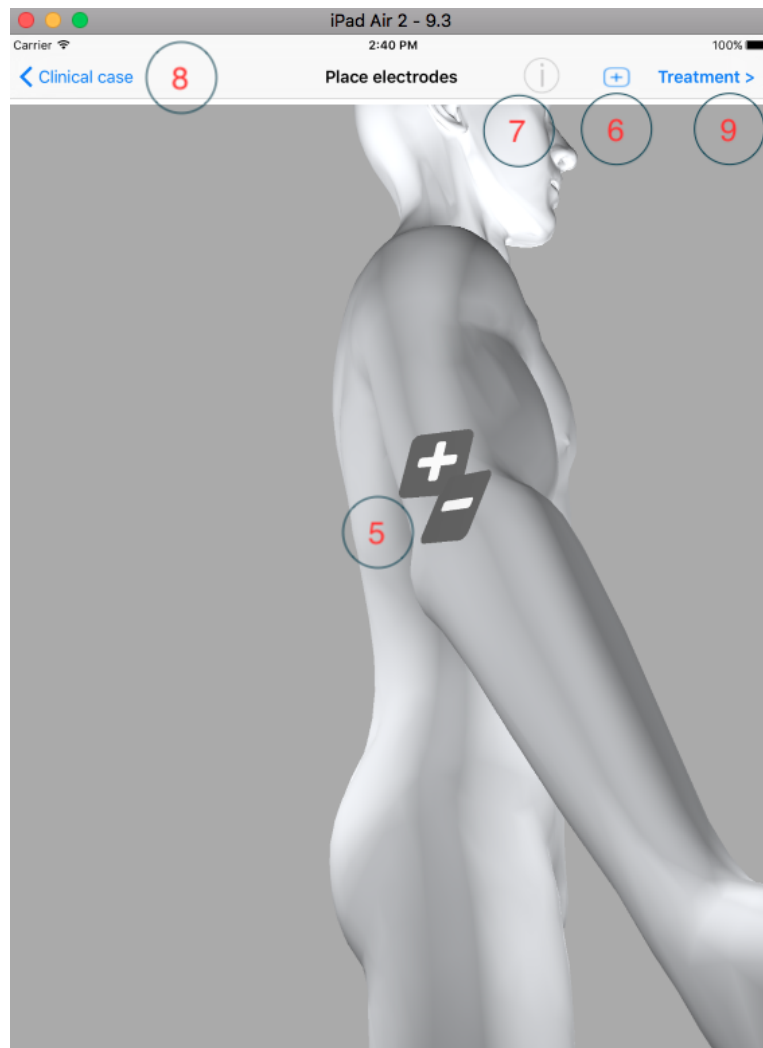


Ilustración 4.12

5 -> Coloca/mueve electrodos.

6 -> Abre la ventana de la configuración de los electrodos. Se puede cambiar la polaridad del electrodo, el número del circuito y el tamaño del electrodo.

7 -> La ventana con la información sobre el tratamiento está desactivado.

8 -> Retrocede a la vista 'caso clínico'.

9 -> Avanza a la vista 'selección familia corriente dispositivo'.

4.4.3 Vista selección familia corriente dispositivo

En esta vista el usuario podrá seleccionar el tipo de familia. En este modo de juego todas las familias de corrientes son seleccionables.



Ilustración 4.13

10 -> Permite subir o bajar la selección.

11 -> Retrocede a la vista 'colocación de electrodos'.

12 -> Avanza a la vista 'selección corriente dispositivo'.

4.4.4 Vista selección corriente dispositivo

Hay algunos tipos de corriente que tienen a su vez tipos de corrientes. En esta vista el usuario podrá seleccionarlás en caso de haberlas.

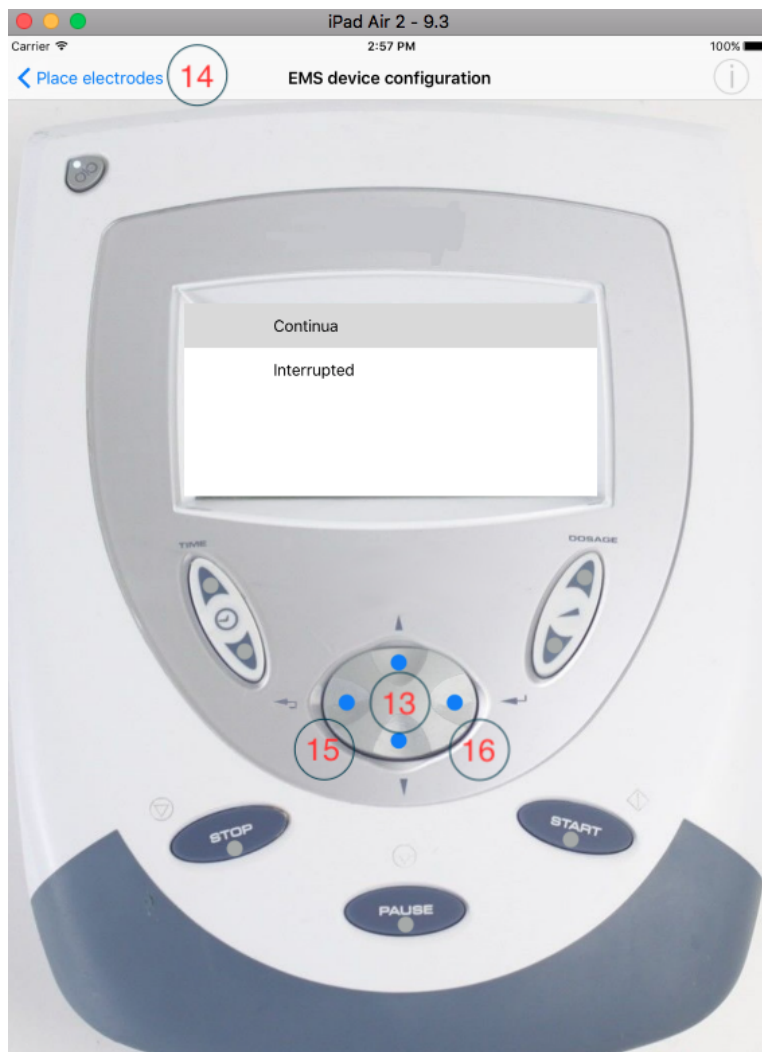


Ilustración 4.14

13 -> Permite subir o bajar la selección.

14 -> Retrocede a la vista 'colocación de electrodos'.

15 -> Retrocede a la vista 'selección familia corriente dispositivo'.

16 -> Avanza a la vista 'configuración dispositivo'.

4.4.5 Vista configuración dispositivo

En esta vista el usuario podrá configurar los parámetros del dispositivo sin restricciones de ningún tipo.



Ilustración 4.15

17 -> Permite subir o bajar la selección.

18 -> Permite cambiar el valor de la selección.

19 -> Retrocede a la vista 'colocación de electrodos.'

20 -> Retrocede a la vista 'selección corriente dispositivo'.

21 -> Avanza a la vista 'inicio dispositivo'.

4.4.6 Vista inicio dispositivo

En esta vista el usuario iniciará el dispositivo pudiendo cambiar los valores de intensidad y tiempo de tratamiento antes o después del inicio del dispositivo. Además, cuando se inicie el dispositivo aparecerá un paciente que se comunicará a través de mensajes de tolerancia al dolor en función de la intensidad de corriente.



Ilustración 4.16

22 -> Permite cambiar el tiempo de tratamiento.

23 -> Permite cambiar el valor de la intensidad de corriente.

24 -> Inicia el dispositivo y aparece el paciente. Se permite además pausar el tiempo y parar el dispositivo.

25 -> Retrocede a la vista 'colocación de electrodos'.

26 -> Retrocede a la vista 'configuración dispositivo'.

4.4.7 Vista evaluación

En esta vista el usuario recibirá una puntuación desglosada en bloques y apartados en la que podrá descubrir sus fallos y aciertos además de una nota final.

ELECTRODE PLACEMENT		
Choice of zone		✓
Method		✓
Technique		✓
Electrodes type		✓
Electrodes polarity		✓
Zone orientation	100 %	✓
Zone distribution	31 %	✗
Zone centrality	26 %	✗
CURRENT CONFIGURATION		
Current		✓
Impulse time	83 %	✓
Frequency	100 %	✓
Polarity		✓
Modulation		✓
Intensity	100 %	✓
Time of treatment	100 %	✓
OVERALL SCORE		
Final score	94 %	✓

27 Finalizar

Ilustración 4.17

27 -> Finaliza el proceso y regresa a la primera vista 'selección modo de juego'.

5 CONCLUSIONES Y TRABAJOS FUTUROS

Gracias a este TFG, pienso que los avances realizados en este proyecto que se inició años atrás son agigantados. Se terminó de desarrollar el modo estudio, el modo evaluación también aunque hay que pulir algunos detalles. Se realizaron pruebas de testeo y cuestionarios de satisfacción por parte de los alumnos.

Me parece un trabajo muy completo en el que he adquirido muchísimos conocimiento, tanto técnicos como de trato con el cliente.

En cuanto a líneas de trabajo futuras. Creo importante profundizar en estas:

- Llegar a un consenso en la evaluación de cada variable.
- Realizar pruebas aún más exhaustivas.
- Seguir realizando cuestionarios de satisfacción y propuesta de mejora a los alumnos. Al fin y al cabo hay que desarrollar este producto en torno a sus necesidades.

6 APÉNDICES

6.1 Zonas corporales

- Cabeza
 - *Cara*
- Miembro superior
 - *Hombro*
 - *Brazo*
 - *Antebrazo*
 - *Muñeca*
 - *Mano*
- Tronco
 - Zona anterior
 - *Región torácica*
 - *Región abdominal*
 - Zona posterior
 - *Trapezio*
 - *Región cervical*
 - *Región dorsal*
 - *Región lumbar*
 - *Región lumbar baja*
 - *Región sacra*
- Miembro inferior
 - *Región glútea*
 - *Cadera*
 - *Muslo*
 - *Rodilla*
 - *Pierna*
 - *Tobillo*
 - *Pie*

6.2 Tablas de corrientes, valores y mensajes

Debido a las dimensiones del mismo, junto a este archivo, se encuentra la tabla de corrientes, valores de parámetros y mensajes del paciente con el nombre “Tabla de corrientes.docx”

6.3 Cuestionario de opinión

Estimado alumno/a, con el fin de evaluar y conocer su opinión en relación a la aplicación de electroterapia que usted ha tenido la oportunidad de utilizar y manejar, se le presentan una serie de cuestiones que rogamos lea detenidamente y conteste según se indica.

El cuestionario pretende evaluar la aplicación como herramienta de evaluación de contenidos principales. Otros aspectos como diseño, manejo... de la misma ya han sido evaluados.

Por favor, marque con una X la respuesta que mejor describa su grado de acuerdo en relación a las siguientes afirmaciones: (Refleje su opinión con toda libertad)

1.-La aplicación permite recordar de forma sencilla y guiada contenidos importantes de la asignatura.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
-----------------------------	------------------	---------	---------------	--------------------------

2.-El aprendizaje guiado de esta aplicación ofrece una retroalimentación muy útil.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
-----------------------------	------------------	---------	---------------	--------------------------

3.-La aplicación me permite practicar aspectos de los contenidos (elección zona del paciente, colocación de electrodos) que no presentan los métodos de aprendizaje convencional.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
-----------------------------	------------------	---------	---------------	--------------------------

4.-La aplicación ayuda al estudio autónomo de la asignatura.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
-----------------------------	------------------	---------	---------------	--------------------------

5.-La aplicación me ayuda a comprobar mi nivel de contenidos aprendidos.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
-----------------------------	------------------	---------	---------------	--------------------------

6.-La aplicación me permite conocer en qué aspectos de la asignatura debo de mejorar

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
-----------------------------	------------------	---------	---------------	--------------------------

7.-Pienso que es un soporte útil como sistema de autoevaluación

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
-----------------------------	------------------	---------	---------------	--------------------------

8.-Pienso que podría ser útil su uso como sistema de evaluación continuada de la asignatura.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
-----------------------------	------------------	---------	---------------	--------------------------

9.-Pienso que podría ser útil su uso como sistema de evaluación final de la asignatura.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
-----------------------------	------------------	---------	---------------	--------------------------

10.-El uso de esta aplicación ha despertado mi interés por las Tecnologías de la Información que pueden utilizarse como método de evaluación.

Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
-----------------------------	------------------	---------	---------------	--------------------------

6.4 Resultados cuestionario de opinión

Debido a las dimensiones del mismo, junto a este archivo, se encuentra el resultado del cuestionario de opinión con el nombre “Resultados cuestionario opinión.docx”

6.5 Guía original del Trabajo Fin de Título

(Cód.: 18/19-2494) Serious game para el aprendizaje de las técnicas de electroterapia orientada a alumnos del Grado en Fisioterapia

Tutor del TFG: **ANTONIO JESÚS RUEDA RUIZ**

Modalidad: Proyecto de Ingeniería | Tipo: TFG Específico

Número máximo de estudiantes: 1 (1 asignados)

Idioma: Castellano

Asignado al alumno con DNI:26511311Q

Segundo tutor del TFG: **CARMEN MARTÍNEZ CRUZ**

Este trabajo fin de grado pretende continuar un trabajo anterior en el que se diseñó una aplicación iOS para facilitar el estudio de las técnicas de electroterapia, orientada a alumnos del Grado en Fisioterapia. El aprendizaje de estas técnicas plantea dificultades a los alumnos debido a la aridez de la materia y al elevado número de corrientes y parámetros físicos que es necesario manejar. En este TFG queremos introducir la gamificación para motivar más al alumno y evaluar de manera precisa su dominio de la asignatura. El serious game que proponemos simulará un tratamiento real de electroterapia, describiendo los síntomas de un paciente que el alumno tendrá que tratar eligiendo un tipo de corriente, configurando los parámetros de dicha corriente y el tiempo de tratamiento, y colocando los correspondientes electrodos en el cuerpo del paciente. Usaremos una evaluación mediante lógica difusa del tratamiento realizado por el alumno para que sea lo más realista y flexible posible, es decir, que se parezca lo máximo posible a la evaluación realizada por el profesor de la asignatura o por un experto en la materia.

Conocimientos Previos

- Programación iOS en Swift
- Gráficos básicos
- Programación de videojuegos (recomendado)
- Lógica difusa (recomendado)

Objetivos del TFG

- Diseñar e implementar un serious game para el aprendizaje de las técnicas de electroterapia.
- Ser capaz de continuar el desarrollo de un proyecto existente de cierta complejidad, adaptando su código y añadiendo nuevas funcionalidades.
- Aprender a programar aplicaciones iOS en el lenguaje Swift.

Metodología a Desarrollar

- Diseñar el esquema general de un serious game para el aprendizaje de las técnicas de electroterapia.
- Proponer métodos de evaluación de las distintas decisiones que debe tomar el alumno: colocación de electrodos, elección de corriente, configuración de intensidad, frecuencia y tiempo de tratamiento, utilizando técnicas de evaluación difusas para permitir mayor flexibilidad.

- Implementar el juego e integrarlo con la aplicación iOS para el aprendizaje de la electroterapia existente.
- Validar el juego con expertos en la materia.
- Probar el juego con alumnos del Grado en Fisioterapia que estén cursando o hayan cursado la asignatura.

Documentos y Formatos de Entrega

- Memoria del trabajo realizado en formato digital, desarrollando los puntos indicados en el apartado de Metodología

7 BIBLIOGRAFÍA

1. Cómo saber si dos líneas intersectan:
<https://www.codeproject.com/Tips/862988/Find-the-Intersection-Point-of-Two-Line-Segments>
2. MVC: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
3. Development and Implementation of a Mobile Application to Improve University Teaching of Electrotherapy Ángeles Díaz-Fernández; Juan-José Jiménez-Delgado; María-Catalina Osuna-Pérez; Antonio-Jesús Rueda-Ruiz; Félix Paulano-Godino, 2016 International Conference on Interactive Mobile Communication, Technologies and Learning (IMCL), San Diego (USA), 17-19 Octubre, 2016, pp. 33-37
4. <http://tauja.ujaen.es/handle/10953.1/5946>