



UNIVERSIDAD DE VALLADOLID



ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA

**Grado en Ingeniería de Tecnologías Específicas de
Telecomunicación. Mención en Telemática**

**Desarrollo de una aplicación móvil
mediante Xamarin para el apoyo a
terapeutas en la rehabilitación cognitiva
de pacientes**

AUTOR: MARCOS MUÑOZ MORALES

TUTORA: MÍRIAM ANTÓN RODRÍGUEZ

Septiembre de 2018





TITULO: DESARROLLO DE UNA APLICACIÓN MÓVIL
MEDIANTE XAMARIN PARA EL APOYO A
TERAPEUTAS EN LA REHABILITACIÓN
COGNITIVA DE PACIENTES

AUTOR: MARCOS MUÑOZ MORALES

TUTORA: MÍRIAM ANTÓN RODRÍGUEZ

DEPARTAMENTO: TEORÍA DE LA SEÑAL, COMUNICACIONES E
INGENIERIA TELEMÁTICA

Miembros del tribunal

PRESIDENTE: MÍRIAM ANTÓN RODRÍGUEZ

SECRETARIO DAVID GONZÁLEZ ORTEGA

VOCAL: MARIO MARTÍNEZ ZARZUELA

SUPLENTE 1: FRANCISCO JAVIER DÍAZ PERNAS

SUPLENTE 2: M.^a ÁNGELES PÉREZ JUÁREZ

FECHA DE LECTURA: SEPTIEMBRE de 2018

CALIFICACIÓN:





RESUMEN:

España es uno de los países con mayor tasa de envejecimiento y sobre-envejecimiento de la unión europea, perspectiva no muy esperanzadora para el futuro de los sistemas públicos de seguridad social. Este aumento de la edad media de la población también se traduce en un aumento de personas con enfermedad mental y otras discapacidades. Si bien es cierto, que hay muchos factores que afectan al deterioro cognitivo como son factores hereditarios, exposición a determinados factores ambientales previos al nacimiento o la misma química del cerebro. Sin embargo, el aumento de edad es un importante factor de riesgo, pasado cierto punto es necesario realizar un seguimiento más severo para controlar el deterioro cognitivo.

Actualmente, el gran crecimiento tecnológico que se produce día a día hace que el mundo del desarrollo de aplicaciones sea uno de los sectores punteros del mercado. Este trabajo fin de grado aborda el aprovechamiento de la tecnología actual para el diseño de una aplicación que ayude a los profesionales en la evaluación y rehabilitación del deterioro cognitivo de pacientes con enfermedades de salud mental.

La herramienta que se propone servirá de complemento a las ya desarrolladas y utilizadas por Fundación INTRAS en sus centros, proporcionando una solución de movilidad para aquellos pacientes que no pueden asistir a las clínicas de Neuropsicología. De esta forma, pueden llevar a cabo sus ejercicios de evaluación y rehabilitación a través del sistema EntrenaTu. El objetivo de este proyecto es que estos pacientes puedan utilizarla sin importar el dispositivo del que dispongan o si disponen o no de conexión a internet. El logro principal es el desarrollo multiplataforma de esta aplicación mediante una tecnología novedosa como es Xamarin. La aplicación se idea para su uso en dispositivos con Windows 10, iPhone, iPad o Smartphones y Tablets Android, es decir, los dispositivos móviles más comunes. Finalmente, cabe destacar su sencilla interfaz gráfica, facilitando la comprensión para personas mayores y con un diseño responsivo a cualquier dispositivo.

PALABRAS CLAVE:

Deterioro Cognitivo

Terapeuta

Serious Games

Xamarin

Dispositivo Móvil



ABSTRACT:

Spain is one of the countries with higher rates of ageing and over-ageing in the European Union, an outlook which is not very encouraging for the future of the public social benefits systems. This increase in the average age of the population also translates into an increase in people with mental illness and other disabilities. It is widely accepted that there are many factors that affect cognitive deterioration, such as hereditary factors, exposure to certain environmental factors prior to birth or the own chemistry of the brain. However, the age increase is also important risk factor, meaning that after a certain point a more severe follow-up is necessary to control cognitive deterioration.

Currently, the great technological growth that occurs every day makes the world of applications development one of the leading sectors of the market. This end-of-degree Project addresses the use of current technology for the design of an application that assist professionals in the evaluation and rehabilitation of cognitive deterioration of patients with mental health illness.

The proposed tool will serve as a complement to those already developed and used by the INTRAS Foundation in their care centers, that also provide of a mobility solution to those patients that can't attend the Neuropsychology clinics. This way, they can carry out their evaluation or rehabilitation exercises through the EntrenaTu system. The goal of this project is that these patients can use it regardless of the device they have available or whether they have an Internet connection or not. The main achievement is the multiplatform development of this application through a new technology such as Xamarin. The application is designed for its use on devices running on Windows 10, iPhone, iPad or Android Smartphones and Tablets, that are, the most common mobile devices. Finally, it is worth highlighting its simple graphical interface, facilitating understanding for older people and with a responsive design to any device.

KEY WORDS:

Cognitive Deterioration

Therapist

Serious Games

Xamarin

Mobile Device



“A mi familia por su plena confianza en mí.

*A mis compañeros y amigos con los que he compartido estos años de
universidad.*

*A Fundación INTRAS por la oportunidad y a todos mis compañeros
haciendo fácil el trabajo del día a día.*

*A mi tutora, Míriam Antón Rodríguez, por la oportunidad y confianza al
inicio de este trabajo.”*





Índice de contenido

1. INTRODUCCIÓN:	15
1.1 FUNDACIÓN INTRAS:	15
1.2 GRADIOR: QUE ES, EVOLUCIÓN Y OBJETIVO	16
1.3 OTROS PROYECTOS: PROYECTO INTERABLE	16
1.4 FASES Y MÉTODOS	17
1.5 MEDIOS	17
1.6 ORGANIZACIÓN DEL DOCUMENTO	18
2. ESTUDIO TECNOLOGÍAS APLICACIONES MÓVILES	19
2.1 APLICACIONES NATIVAS	21
2.1.1 <i>Android</i>	22
2.1.2 <i>iOS</i>	22
2.1.3 <i>Windows</i>	23
2.2 APLICACIONES WEB	24
2.3 APLICACIONES HÍBRIDAS	26
2.4 APLICACIONES GENERADAS	27
2.5 SOLUCIÓN ESCOGIDA Y JUSTIFICACIÓN	28
3. XAMARIN	31
3.1 XAMARIN.FORMS	31
4. ANÁLISIS	37
4.1 ACTORES DEL SISTEMA	37
4.2 DIAGRAMA DE CASOS DE USO	37
4.3 ESPECIFICACIONES DE LOS CASOS DE USO	38
4.3.1 <i>Lanzar tratamientos</i>	38
4.3.2 <i>Descargar elementos y ejercicios</i>	40
4.3.3 <i>Salir de la aplicación</i>	41
4.3.4 <i>Asignar paciente activo</i>	42
4.3.5 <i>Borrar paciente activo</i>	43
4.3.6 <i>Iniciar sesión terapeuta</i>	44
4.3.7 <i>Comprobar conexión</i>	45
4.3.8 <i>Realizar consultas</i>	46
4.3.9 <i>Obtener actualización de tratamiento</i>	47
4.4 REQUISITOS NO FUNCIONALES	48
4.5 REQUISITOS DE INFORMACIÓN	49
5. MANUAL DE USUARIO	51
5.1 ICONO DE LA APLICACIÓN	51
5.2 MODALIDADES	51
5.2.1 <i>Atención</i>	51
5.2.2 <i>Orientación</i>	52
5.2.3 <i>Calculo</i>	52
5.2.4 <i>Razonamiento</i>	53
5.2.5 <i>Percepción</i>	53
5.2.6 <i>Lenguaje</i>	54
5.2.7 <i>Función ejecutiva</i>	54
5.2.8 <i>Memoria</i>	55
5.3 MÉTODO CREACIÓN PANTALLA DE EJERCICIOS UNIVERSAL	57
5.4 PANTALLAS	58
5.4.1 <i>Pantalla de descarga de contenido</i>	58
5.4.2 <i>Pantalla inicio paciente</i>	58
5.4.3 <i>Pantalla autenticación terapeuta</i>	59
5.4.4 <i>Pantalla de asignación de paciente activo</i>	60



5.4.5	Pantalla de estadísticas	60
5.4.6	Pantalla de introducción	61
5.4.7	Distraction screen	62
5.4.8	Atención Selectiva Visual Execution Screen	63
5.4.9	Atención vigilancia Color Execution Screen	64
5.4.10	Ejecutiva cambio reglas si no execution screen	65
5.4.11	Ejecutiva tarea claves sequential execution screen	65
5.4.12	Puzzle execution screen	66
5.4.13	Ejecutiva ordenación execution screen	66
5.4.14	Ejecutiva tarea números y letras presentation screen	67
5.4.15	Span sequential presentation Screen	67
5.4.16	Ejecutiva interferencias simultaneous execution screen	68
5.4.17	Ejecutiva interferencia si no execution screen	68
5.4.18	Ejecutiva inhibición auditiva execution screen	69
5.4.19	Ejecutiva inhibición execution screen	69
5.4.20	Solución problemas presentational screen	70
5.4.21	Solución problemas execution screen	70
5.4.22	Solución problemas alternative screen	71
5.4.23	Ejecutiva tarea contar presentation screen	72
5.4.24	Ejecutiva tarea contar execution screen	72
5.4.25	Ejecutiva tarea contar alternative screen	72
5.4.26	Memoria simultaneous execution screen	73
5.4.27	Memoria sequential execution screen	73
5.4.28	Implicit execution screen	74
5.4.29	Secuencias execution screen	75
5.4.30	Percepción color execution screen	75
5.4.31	Percepción visual execution screen	76
5.4.32	Razonamiento ordenar execution screen	76
5.4.33	Simultaneous presentation screen	77
5.4.34	Simultaneous execution screen	77
5.4.35	Momento día execution screen	78
5.4.36	Year execution screen	78
5.4.37	Estaciones 4 execution screen	79
5.4.38	Día semana execution screen	80
5.4.39	Día mes execution screen	80
6.	CONCLUSIONES	83
7.	LÍNEAS FUTURAS	85
8.	BIBLIOGRAFÍA	87
ANEXO 1: VISUAL STUDIO		89
	REQUISITOS MÍNIMOS	89
	INSTALACIÓN	90
	CREACIÓN DE PRIMER PROYECTO XAMARIN FORMS	91
ANEXO 2: VISUAL STUDIO APP CENTER		95



Índice de ilustraciones

ILUSTRACIÓN 1: PIRÁMIDE DE POBLACIÓN DE ESPAÑA (BANKINTER, 2015).....	15
ILUSTRACIÓN 2: EJEMPLO PRUEBA INTERABLE.....	17
ILUSTRACIÓN 3: EVOLUCIÓN APARIENCIA DESDE TELÉFONO MÓVIL HASTA SMARTPHONE.....	19
ILUSTRACIÓN 4: DISPOSITIVOS MÓVILES (SMARTPHONE + TABLETS) VS EQUIPOS DE ESCRITORIO.....	20
ILUSTRACIÓN 5: TASA DE PENETRACIÓN DE LOS SISTEMAS OPERATIVOS PARA SMARTPHONE EN ESPAÑA DESDE 2014 HASTA 2016.....	21
ILUSTRACIÓN 6: FAMILIA DE DISPOSITIVOS WINDOWS UWP.....	24
ILUSTRACIÓN 7: LOGOS PHONEGAP Y APACHE CORDOVA.....	26
ILUSTRACIÓN 8: DESARROLLO SOFTWARE APLICACIONES GENERADAS.....	27
ILUSTRACIÓN 9: EJEMPLO DE DISEÑO INTERFAZ C#.....	32
ILUSTRACIÓN 10: EJEMPLO INTERFAZ MEDIANTE XAML.....	33
ILUSTRACIÓN 11: INTERFAZ GRÁFICA RESULTANTE CON AMBOS CÓDIGOS.....	33
ILUSTRACIÓN 12: ARQUITECTURA PROYECTOS PCL Y SHARED PROJECTS.....	34
ILUSTRACIÓN 13: EJEMPLO PROYECTO PCL.....	34
ILUSTRACIÓN 14: EJEMPLO SHARED PROJECTS.....	35
ILUSTRACIÓN 15: EJEMPLO DE COMPILACIÓN CONDICIONAL.....	35
ILUSTRACIÓN 16: DIAGRAMA DE CASOS DE USO.....	38
ILUSTRACIÓN 17: DIAGRAMA DE FLUJO LANZAR TRATAMIENTO.....	40
ILUSTRACIÓN 18: ICONO APLICACIÓN ENTRENATU.....	51
ILUSTRACIÓN 19: EJEMPLO FICHERO XML EJERCICIO.....	57
ILUSTRACIÓN 20: COMANDOS DESERIALIZACIÓN XML EJERCICIO.....	58
ILUSTRACIÓN 21: PANTALLA DE DESCARGA DE CONTENIDO.....	58
ILUSTRACIÓN 22: PANTALLA INICIAL SIN PACIENTE ASIGNADO.....	59
ILUSTRACIÓN 23: PANTALLA INICIAL CON PACIENTE ASIGNADO.....	59
ILUSTRACIÓN 24: PANTALLA AUTENTICACIÓN TERAPEUTA.....	60
ILUSTRACIÓN 25: PANTALLA ASIGNACIÓN DE PACIENTE ACTIVO.....	60
ILUSTRACIÓN 26: PANTALLA DE ESTADÍSTICAS.....	61
ILUSTRACIÓN 27: INTRODUCTION SCREEN.....	62
ILUSTRACIÓN 28: EJEMPLO PANTALLA DE INTRODUCCIÓN EN CASO DE FALLO O ACIERTO DE UN EJERCICIO CON VARIAS SECUENCIAS.....	62
ILUSTRACIÓN 29: DISTRACTION SCREEN.....	63
ILUSTRACIÓN 30: ATENCIÓN SELECTIVA VISUAL EXECUTION SCREEN.....	63
ILUSTRACIÓN 31: ATENCIÓN VIGILANCIA COLOR O ATENCIÓN SOSTENIDA COLOR.....	64
ILUSTRACIÓN 32: ATENCIÓN VIGILANCIA DESTELLOS O ATENCIÓN SOSTENIDA DESTELLOS.....	64
ILUSTRACIÓN 33: EJECUTIVA CAMBIO REGLAS SI NO EXECUTION SCREEN.....	65
ILUSTRACIÓN 34: EJECUTIVA TAREA CLAVES SEQUENTIAL EXECUTION SCREEN.....	66
ILUSTRACIÓN 35: PUZZLE EXECUTION SCREEN.....	66
ILUSTRACIÓN 36: EJECUTIVA ORDENACIÓN EXECUTION SCREEN.....	67
ILUSTRACIÓN 37: EJECUTIVA TAREA NÚMEROS Y LETRAS PRESENTATION SCREEN.....	67
ILUSTRACIÓN 38: SPAN SEQUENTIAL PRESENTATION SCREEN.....	68
ILUSTRACIÓN 39: EJECUTIVA INTERFERENCIAS SIMULTANEOUS EXECUTION SCREEN.....	68
ILUSTRACIÓN 40: EJECUTIVA INTERFERENCIA SI NO EXECUTION SCREEN.....	69
ILUSTRACIÓN 41: EJECUTIVA INHIBICIÓN AUDITIVA EXECUTION SCREEN.....	69
ILUSTRACIÓN 42: EJECUTIVA INHIBICIÓN EXECUTION SCREEN.....	70
ILUSTRACIÓN 43: SOLUCIÓN PROBLEMAS PRESENTATIONAL SCREEN.....	70
ILUSTRACIÓN 44: SOLUCIÓN PROBLEMAS EXECUTION SCREEN.....	71
ILUSTRACIÓN 45: SOLUCIÓN PROBLEMAS EXECUTION SCREEN.....	71
ILUSTRACIÓN 46: EJECUTIVA TAREA CONTAR PRESENTATION SCREEN.....	72
ILUSTRACIÓN 47: EJECUTIVA TAREA CONTAR ALTERNATIVE SCREEN.....	73
ILUSTRACIÓN 48: MEMORIA SIMULTANEOUS EXECUTION SCREEN.....	73
ILUSTRACIÓN 49: MEMORIA SEQUENTIAL EXECUTION SCREEN.....	74
ILUSTRACIÓN 50: IMPLICIT EXECUTION SCREEN.....	74
ILUSTRACIÓN 51: SECUENCES EXECUTION SCREEN.....	75
ILUSTRACIÓN 52: PERCEPCIÓN COLOR EXECUTION SCREEN.....	76
ILUSTRACIÓN 53: PERCEPCIÓN VISUAL EXECUTION SCREEN.....	76



ILUSTRACIÓN 54: RAZONAMIENTO ORDENAR EXECUTION SCREEN	77
ILUSTRACIÓN 55: SIMULTANEOUS PRESENTATION SCREEN	77
ILUSTRACIÓN 56: SIMULTANEOUS EXECUTION SCREEN	78
ILUSTRACIÓN 57: MOMENTO DÍA EXECUTION SCREEN	78
ILUSTRACIÓN 58: YEAR EXECUTION SCREEN	79
ILUSTRACIÓN 59: ESTACIONES 4 EXECUTION SCREEN	79
ILUSTRACIÓN 60: DÍA SEMANA EXECUTION SCREEN	80
ILUSTRACIÓN 61: DÍA MES EXECUTION SCREEN.....	81
ILUSTRACIÓN 62: DÍA MES EXECUTION SCREEN SOLUCIÓN FINAL	81
ILUSTRACIÓN 63: VERSIONES VISUAL STUDIO	90
ILUSTRACIÓN 64: PAQUETES NECESARIOS DESARROLLO XAMARIN	90
ILUSTRACIÓN 65: ELECCIÓN DE PROYECTO XAMARIN FORMS	91
ILUSTRACIÓN 66: TIPO DE SOLUCIÓN XAMARIN FORMS	92
ILUSTRACIÓN 67: DIRECTORIOS SOLUCIÓN XAMARIN FORMS	92
ILUSTRACIÓN 68: PAQUETES APP CENTER.....	95
ILUSTRACIÓN 69: REGISTRO DE ANALÍTICAS APP CENTER.....	95
ILUSTRACIÓN 70: CÓDIGO CONTROL EXCEPCIONES CONTROLADAS APP CENTER.....	96
ILUSTRACIÓN 71: VISUALIZACIÓN EXCEPCIONES APP CENTER	96
ILUSTRACIÓN 72: VISUALIZACIÓN DE UNA EXCEPCIÓN APP CENTER	96



Índice de tablas

TABLA 1: ESPECIFICACIONES CASO DE USO LANZAR TRATAMIENTO.....	39
TABLA 2: ESPECIFICACIONES CASO DE USO DESCARGAR ELEMENTOS Y EJERCICIOS	41
TABLA 3: ESPECIFICACIONES CASO DE USO SALIR DE LA APLICACIÓN	42
TABLA 4: ESPECIFICACIONES CASO DE USO ASIGNAR PACIENTE ACTIVO	43
TABLA 5: ESPECIFICACIONES CASO DE USO BORRAR PACIENTE ACTIVO	44
TABLA 6: ESPECIFICACIONES CASO DE USO INICIAR SESIÓN TERAPEUTA	45
TABLA 7: ESPECIFICACIONES CASO DE USO COMPROBAR CONEXIÓN	46
TABLA 8: ESPECIFICACIONES CASO DE USO REALIZAR CONSULTAS	47
TABLA 9: ESPECIFICACIONES CASO DE USO OBTENER ACTUALIZACIÓN DE TRATAMIENTO	48



1. INTRODUCCIÓN:

El gran avance tecnológico surgido en los últimos años ha permitido complementar a otros sectores, como ha sido el caso en la medicina. Hoy en día, diferentes centros médicos utilizan la creación de aplicaciones como ayuda a tratamientos. Estas aplicaciones permiten la consulta tanto del paciente como del facultativo. Dando una ayuda extra al paciente, y mientras tanto, el doctor recibe la información en tiempo real sin llegar a ser necesario un encuentro presencial.

El aumento de la esperanza de vida ha traído consigo un envejecimiento de la población en los países más desarrollados y esta no hace más que aumentar como vemos en la siguiente ilustración. En ella puede apreciarse la población actual de cada rango de edad así como las estimaciones de futuro (Bankinter, 2015).

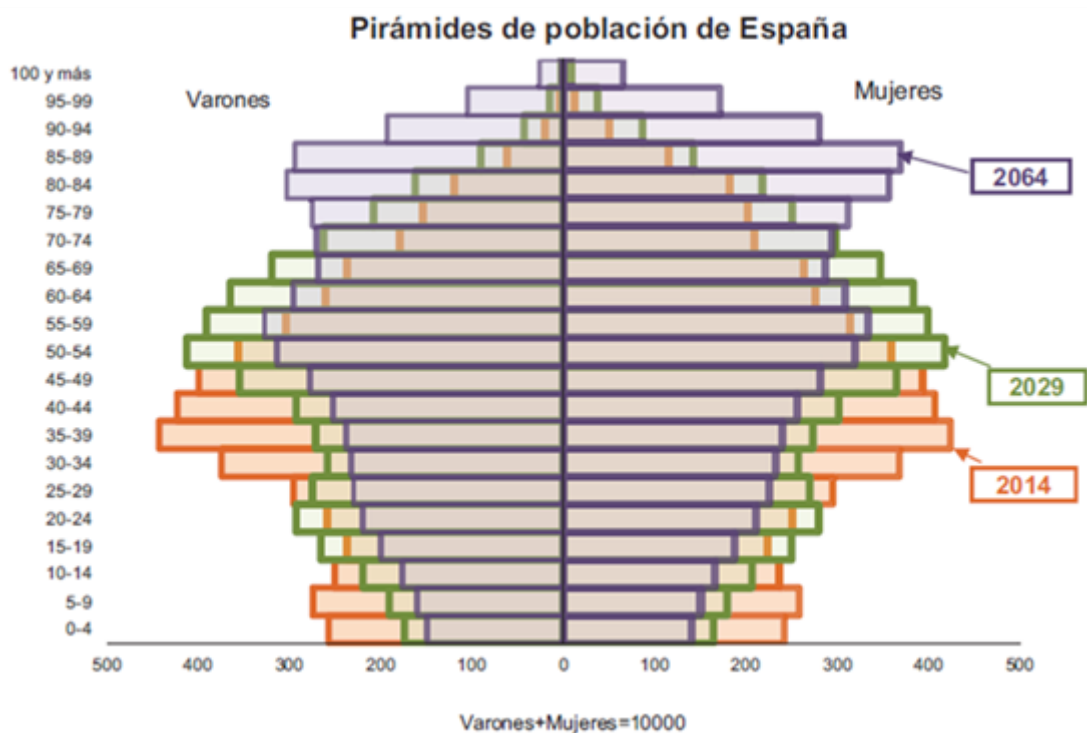


Ilustración 1: Pirámide de población de España (Bankinter, 2015).

Este aumento en la edad de la población trae consigo ese deterioro cognitivo, que en la mayoría de los casos, es provocado por la edad. Desde síntomas leves como puede ser olvidar donde se dejan las llaves, hasta en casos más avanzados, suponer un riesgo para la vida diaria del afectado.

Aquí es donde entra en juego el uso de nuevas tecnologías como respaldo hacia los pacientes y personal sanitario o socio sanitario, pudiendo llegar a llevar unos servicios a personas que no tienen la capacidad de movilidad para ir a una clínica por ejemplo. Teniendo en cuenta que ese ahorro en desplazamiento no solo es beneficioso para el paciente, sino también para la clínica que lo trata (El Mundo, 2012).

1.1 FUNDACIÓN INTRAS:

Fundación INTRAS trabaja en el ámbito de la recuperación del proyecto vital por personas afectadas por una discapacidad. Desde el diagnóstico y evaluación de la discapacidad por enfermedad mental de la que se trate, hasta su posterior tratamiento y



rehabilitación. Es por ello que Fundación INTRAS centra su atención en personas con discapacidad por enfermedad mental grave y prolongada (INTRAS ,2015).

Se llega a proporcionar alojamiento, apoyo y seguimiento a esta comunidad de modo que sea posible la recuperación de las habilidades funcionales para la autonomía y la prevención del deterioro y recaídas.

En este caso tras Fundación INTRAS se encuentra IDES (I MAS D Y EMPLEO SERVICONSULTING S.L.), creada para potenciar la actividad I+D+I de la fundación. Desarrollando, comercializando y explotando productos y servicios tecnológicos en el ámbito socio sanitario. Es decir, Fundación INTRAS se encarga de la atención directa con el usuario final y por otro lado IDES trabaja en el desarrollo, investigación e innovación tecnológica que hay detrás. Uno de los proyectos más avanzados es el Software de Rehabilitación Neuropsicológica del Deterioro Cognitivo (GRADIOR (GRADIOR, 2017)).

1.2 GRADIOR: QUE ES, EVOLUCIÓN Y OBJETIVO

GRADIOR proporciona apoyo a profesionales en el ámbito de la evaluación y rehabilitación cognitiva; Incluye más de 40000 ejercicios dinámicos divididos en modalidades (atención, percepción, memoria, cálculo, lenguaje y razonamiento) así como nuevos ejercicios con contenidos de la vida real (GRADIOR, 2017).

Aquí es donde entra la tarea de este proyecto, partiendo de este software GRADIOR, una base ya desarrollada para trabajar en un entorno Windows, con una serie de ejercicios desarrollados para evaluación y rehabilitación de pacientes. En este caso se desea desarrollar una herramienta que soporte una movilidad que solo permiten los nuevos dispositivos móviles (Smartphones y Tablets), siendo igualmente compatible con equipos de sobremesa. Llegando a diferentes familias como son los ordenadores de escritorio Windows, Tablets o Smartphone Android o incluso iPads o iPhones con iOS. Todo ello sin modificar o intentando modificar lo menos posible la apariencia, comportamiento y desarrollo de los ejercicios existentes, de modo que tanto el nuevo software como el ya existente sean compatibles.

1.3 OTROS PROYECTOS: PROYECTO INTERABLE

En el proyecto INTERABLE se desarrolló una plataforma con la que combinar la intervención física y cognitiva. Estas áreas son un factor reconocido hacia la protección de las funciones cerebrales tanto en estados de envejecimiento cerebral como en otros estudios del deterioro (INTERABLE, 2017).

Únicamente es necesario un ordenador, una cámara Kinect y la plataforma INTERABLE. Para la intervención física se realiza un entrenamiento con un patrón de ejercicios físicos controlados (ejercicios aeróbicos, de fuerza, equilibrio y flexibilidad, combinando modernas técnicas de captura de movimiento a través del dispositivo Kinect), con diferentes niveles de dificultad. De este modo, el terapeuta, personalizara el plan de ejercicios para que se adapte al usuario. Esta terapia física se combina con ejercicios de memoria y otras funciones cognitivas a fin de evaluar ambas áreas.



Ilustración 2: Ejemplo prueba INTERABLE

1.4 FASES Y MÉTODOS

Para conseguir los objetivos propuestos, se han seguido los siguientes pasos en el desarrollo del proyecto:

- Documentación y razonamiento. Comparativa de las diferentes tecnologías para el desarrollo de una aplicación móvil multiplataforma.
- Estudio de Xamarin. Se estudian las posibilidades de este entorno de desarrollo para la creación de aplicaciones nativas en cada plataforma móvil.
- Desarrollo de la nueva aplicación. Programación de la aplicación, la lógica de negocio debe seguir la línea de la aplicación GRADIOR ya desarrollada para equipos de escritorio con Windows. Diseño de las interfaces para los nuevos dispositivos de cara a la visión correcta por parte del usuario.
- Pruebas. Realización de pruebas tanto por parte del desarrollador como de terapeutas de la clínica, realizando las correcciones necesarias a los posibles fallos que surgieran en la fase de desarrollo.
- Conclusiones y líneas futuras para la mejora y ampliación de la aplicación.

1.5 MEDIOS

Para el desarrollo de la aplicación se consulta habitualmente la página web para desarrolladores de Microsoft, <https://msdn.microsoft.com>, en ella encontramos documentación a cerca de la gran mayoría de clases utilizadas en el entorno de Visual Studio. A su vez también se han consultado manuales, libros y fotos donde se muestran ejemplos y resolución de problemas compatibles con nuestra aplicación.

Para el desarrollo y depuración de la aplicación se han utilizado los siguientes equipos y componentes software:

- Visual Studio 2017.



- Equipo de sobremesa, Windows 10 como sistema operativo, procesador Intel Core I5, 16 GB de memoria RAM, disco duro SSD de 256 GB y disco duro HDD de 1 TB. Equipos Windows 10 pertenecientes a la clínica de GRADIOR para la fase de pruebas.
- Sony Xperia Z3 compact con sistema operativo Android 6.0.1, otros dispositivos con Android 4.0 o superior como pueden ser diferentes Tablets para la prueba en diferentes resoluciones y pantallas.
- Apple iPad 2017 combinado con un iMac para el despliegue en dicho iPad.

1.6 ORGANIZACIÓN DEL DOCUMENTO

Para poder abarcar estos objetivos y seguir su desarrollo de manera fluida, este informe se ha dividido en 9 capítulos.

El primer capítulo, en el cual nos encontramos, se realiza una introducción a los contenidos del trabajo y se contextualiza la finalidad y los objetivos.

En el capítulo dos, se definen diferentes opciones a la hora de desarrollar una aplicación móvil multiplataforma, indicando sus ventajas e inconvenientes. Desarrollando brevemente el porqué de la opción escogida.

Seguidamente, en el capítulo tres, se explican las características principales del entorno de trabajo escogido.

En el cuarto capítulo, se realiza un análisis software de la aplicación, se hace uso de diagramas de caso de uso, o tablas donde detallar las especificaciones de estos.

A continuación, en el capítulo cinco, se realiza un pequeño manual de usuario, este contiene una explicación breve del icono, una descripción de cada modalidad y sub-modalidad de ejercicios disponibles, y una descripción detallada de cada tipo de pantalla con la que es posible encontrarse en una sesión de cualquier tratamiento.

Finalmente, en los últimos capítulos tres últimos capítulos, se detallan las conclusiones, las posibles líneas futuras con las que podría continuarse esta aplicación y toda la bibliografía utilizada en el desarrollo del presente proyecto.

2. ESTUDIO TECNOLOGÍAS APLICACIONES MÓVILES

La enorme evolución desarrollada por parte de los dispositivos móviles ha posibilitado un sinnúmero de nuevas aplicaciones. Muchas de ellas enfocadas hacia necesidades existentes y cotidianas del día a día, aunque otras muchas hacia la creación de necesidades, las cuales, no se tenían en cuenta hasta el momento.

Con los primeros teléfonos móviles, no confundir con dispositivos móviles, la única y exclusiva necesidad que se pretendía satisfacer era la de la comunicación, eran terminales básicos, con alguna aplicación sencilla preinstalada sin posibilidad de modificar, permitiendo personalizar unos mínimos ajustes. Conforme esa tecnología se abarataba los componentes mejoraban y de este modo así lo hacían las capacidades de estos dispositivos. Esto propiciaba que el nivel adquisitivo necesario para adquirir uno de estos terminales se redujera con el tiempo y a su vez estas inversiones permitían a la industria mantener un presupuesto con el que investigar y hacer avanzar esta tecnología.

De únicamente permitir el recibir o realizar llamadas pasamos al añadido de mensajes de texto, pantallas a color, cámara de fotos, redes de datos, etc... El cambio al concepto de teléfono inteligente, donde llego el punto de inflexión. Estos dispositivos habían dejado de tener como función principal la de comunicarse sustituyéndose por la que el usuario dueño del dispositivo quisiera darle, manteniendo las capacidades de los primeros móviles, pero mejorando en muchísimos otros aspectos, y todo por un precio muy inferior al de aquellos primeros terminales. En la siguiente ilustración se muestra la evolución en términos de apariencia, desde los primeros teléfonos móviles a los actuales Smartphones, donde se nota claramente el cambio de tendencia. De buscar un minimalismo en cuanto a tamaño a la aparición de grandes pantallas buscando un equilibrio entre ese tamaño ya no tan reducido y un cómodo manejo.

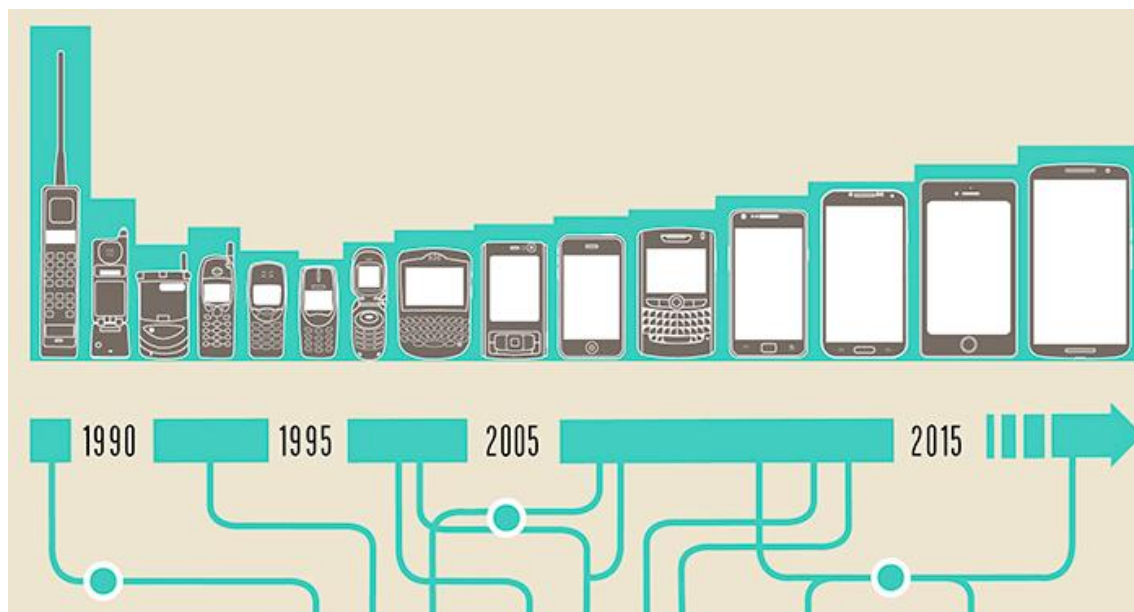


Ilustración 3: Evolución apariencia desde teléfono móvil hasta Smartphone

Si se hace referencia a la definición estricta de dispositivo digital móvil, podemos recurrir a diferentes libros donde se distingue entre esa pequeña diferencia como es la de móvil y fijo. Más en concreto, Esteban Vázquez-Cano y María Luisa Sevillano García, en el libro Dispositivos digitales móviles en educación, definen un dispositivo digital móvil como:

“Un dispositivo digital móvil es un mecanismo diseñado para realizar la función para la que ha sido diseñado, con independencia de la existencia o no de alimentación eléctrica. Es decir, este tipo de dispositivo cuenta con un almacén energético ubicado en la batería (interna o externa) que mantiene, durante un espacio de tiempo, el flujo de electricidad que el dispositivo necesita para realizar la tarea que tiene encomendada. Un ejemplo de dispositivo digital móvil se encuentra en el ordenador portátil o laptop. Un dispositivo digital móvil atendiendo, exclusivamente, al criterio de portabilidad es más versátil que un dispositivo digital fijo.

El dispositivo digital móvil puede funcionar como un dispositivo digital fijo, es decir, conectado a la corriente eléctrica por cable mientras que un dispositivo digital fijo nunca puede funcionar como un dispositivo digital móvil por su incapacidad para mantener, por sí mismo, la alimentación eléctrica.” (Vázquez-Cano & Sevillano García ,2015)

En ningún momento se hará referencia a la posible diferencia en términos de prestaciones de cualquier grado entre dispositivos fijos y móviles. Se ha roto esa barrera que impedía a un dispositivo portátil ser tanto o más potente que un dispositivo fijo en cuanto a cualquiera de sus capacidades.

Para comprobar la evolución del número de dispositivos móviles y fijos se puede hacer uso de diferentes herramientas de estadística. En ellas puede verse la tendencia en el crecimiento del porcentaje de dispositivos móviles y con ello la tendencia en la reducción de cuota de mercado de dispositivos fijos.

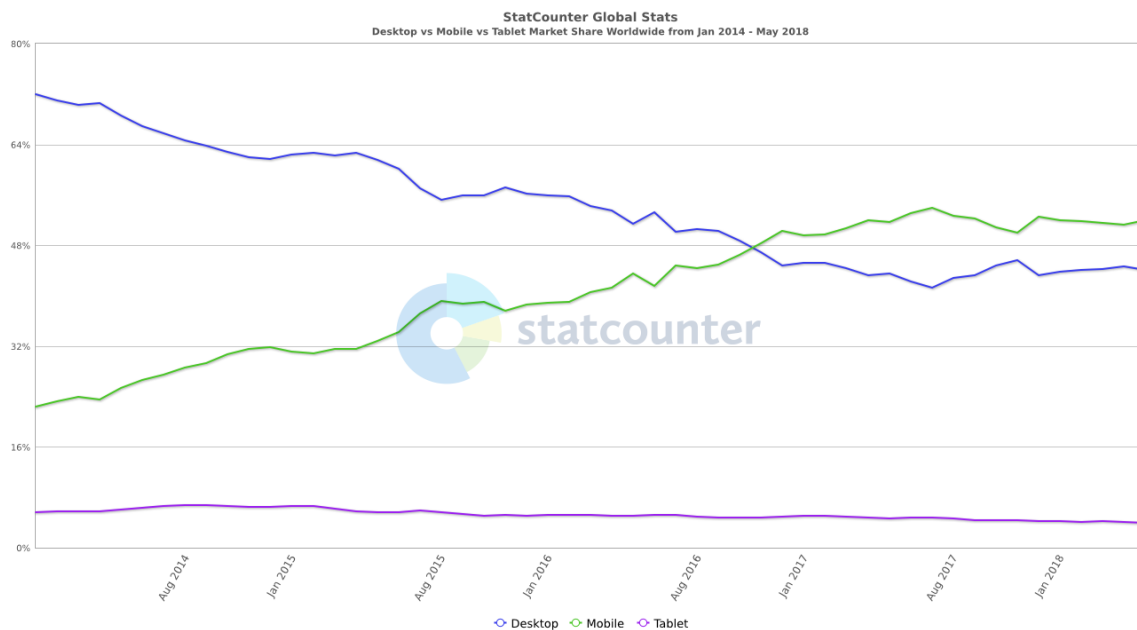


Ilustración 4: Dispositivos móviles (Smartphone + Tablets) vs Equipos de escritorio

No se puede decir que el número de equipos de escritorio se haya reducido, si no que el número de dispositivos móviles ha sufrido un enorme crecimiento global (Statcounter, 2018).

Ahora bien, como se ha comentado al inicio, no se debe confundir teléfono móvil o Smartphone con dispositivo móvil. Gracias a la definición estricta de dispositivo digital móvil puede verse que estos teléfonos móviles o los nuevos Smartphones son un tipo de

dispositivo móvil, al igual que las Tablets siendo estos últimos (Smartphone y Tablet) los grandes dominadores del mercado. Al igual que se hace referencia a los tipos de dispositivos móviles más importantes, también debe hacerse a las compañías dominadoras de este sector, hacia las cuales están diseñadas casi la totalidad de aplicaciones móviles que se desarrollan. En la siguiente ilustración puede verse la tasa de penetración de los sistemas operativos de Smartphone en España (Statista, 2016).

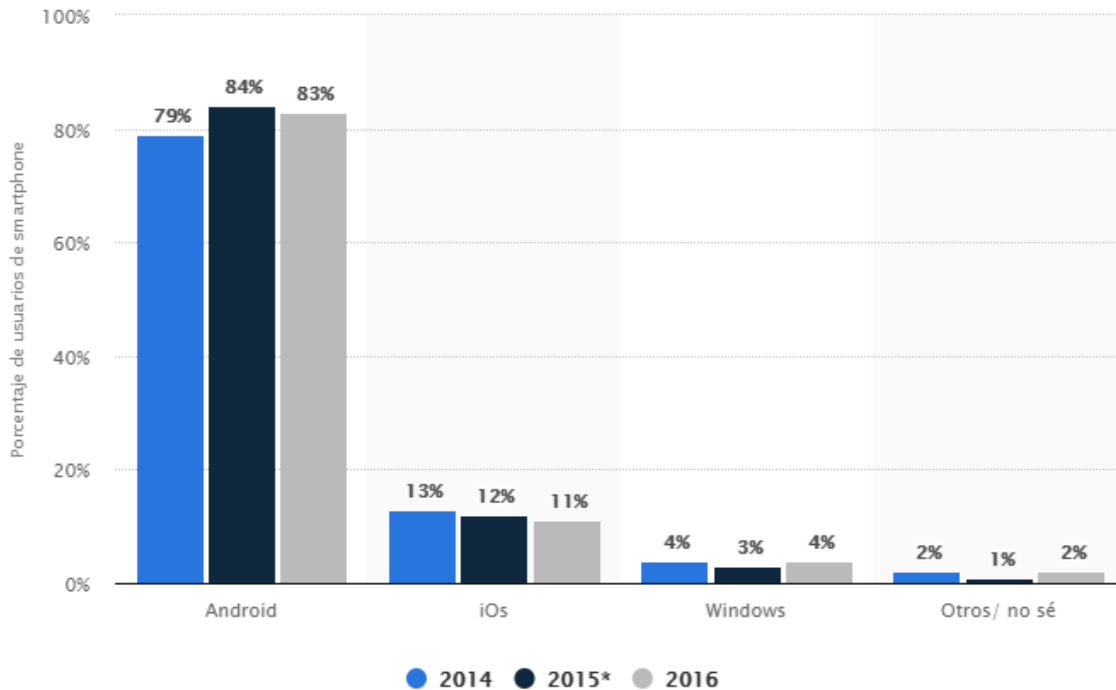


Ilustración 5: Tasa de penetración de los sistemas operativos para Smartphone en España desde 2014 hasta 2016

Estas estadísticas son un factor importante, el deseo es que una aplicación llegue al máximo número de dispositivos y a los sistemas operativos principales de cada uno de ellos. Para cumplir ese fin, existen varios tipos de soluciones que permiten crear diferentes tipos de aplicaciones móviles. Existen 4 tipos de aplicaciones para su utilización en Smartphones y Tablets como son:

- Aplicaciones nativas.
- Aplicaciones web.
- Aplicaciones híbridas.
- Aplicaciones generadas.

A continuación, se recorrerán todas en búsqueda del tipo más conveniente para nuestro proyecto, cada aplicación tendrá sus características y limitaciones, el tipo de aplicación que elijamos condicionará su diseño.

2.1 APLICACIONES NATIVAS

Estas aplicaciones nativas están únicamente desarrolladas con el software que ofrece cada sistema operativo a los programadores, generalmente llamado SDK (Software Development Kit, kit de desarrollo software). En este concepto es el que radica la mayor y más significativa diferencia, cada sistema operativo tiene su propio SDK, lo que



conlleva un desarrollo independiente para cada plataforma, es decir, cada aplicación es programada para un sistema operativo específico.

En términos de rendimiento son superiores a cualquier tipo de aplicación, el utilizar directamente el software específico de cada plataforma hace que pueda expresarse al máximo sus capacidades sin necesitar de estar conectados a ningún servidor externo ni hacer uso de la conexión a internet en ninguno de los casos, a no ser que la propia aplicación lo necesite. La apariencia también será un punto a favor, al utilizar los elementos propios de cada sistema operativo las aplicaciones tendrán una mayor coherencia y consistencia con el resto de las aplicaciones del dispositivo (Cuello et al, 2013).

Un gran inconveniente a tener en cuenta en el caso de pretender crear aplicaciones multiplataforma es el que radica en la primera característica que se ha mencionado, un desarrollo independiente por cada plataforma. Si por ejemplo estamos buscando crear una aplicación para los tres sistemas operativos más demandados las cuentas serán fáciles, deberemos invertir aproximadamente el triple de tiempo que para una aplicación en una única plataforma. Si este es el desarrollo que se desea, se aportaran unos breves conocimientos acerca de los requisitos de cada plataforma.

2.1.1 Android

Se trata de un sistema operativo basado en Linux siguiendo un modelo de código abierto. Es el sistema operativo líder en el mercado de Smartphones y Tablets, lanzando su primera versión en el año 2008 llegando en su última versión a Android 8.0 Oreo (Hébuterne, 2016).

Algunas de sus ventajas y desventajas son las siguientes.

Ventajas:

- Modelo de código abierto, cualquier persona puede realizar una aplicación para Android.
- Lenguaje de programación JAVA, ampliamente extendido, únicamente será necesario instalar su JDK (Java Development Kit, kit de desarrollo Java) el cual es posible instalarlo junto al entorno de desarrollo propio de Google, Android Studio. Dando la posibilidad de que cualquier usuario pueda desarrollar sus aplicaciones y mediante una licencia muy accesible pueda añadirlas a la tienda de aplicaciones.

Desventajas:

- El gran número de dispositivos y de variantes de sistema operativo hace que pocos terminales reciban soporte durante un largo periodo de tiempo.
- La posibilidad de multitarea conlleva un alto consumo de batería.

2.1.2 iOS

Sistema operativo propiedad de Apple Inc. Originalmente desarrollado para el primer iPhone por ello su antecesor se llamaba iPhone OS, posteriormente adaptado para el iPod touch y el iPad. Se reveló su existencia por primera vez en 2007, en la actualidad ya está por su versión iOS 11 (Arias et al, 2016).



Para desarrollar aplicaciones en este sistema operativo es indispensable trabajar con el IDE Xcode dentro de una máquina con sistema operativo MacOS, por lo que ya se impone la obligación de contar con un equipo ya sea sobremesa o portátil de Apple. Se tienen dos posibilidades de lenguaje con los que trabajar:

- Objective-C, extensión orientada a objetos del lenguaje C.
- Swift, más amigable que Objective-C, permite trabajar junto a este.

Una de sus desventajas de cara al desarrollo para iOS es la obligación de contar con una cuenta de desarrollador iOS de Apple, con ella podremos acceder a todas sus ventajas a través de una suscripción de pago anual, desde proyectos de ejemplo hasta manuales de ayuda a parte de poder unirnos a la gran comunidad de desarrolladores de Apple (Arias et al, 2016).

Al igual que con todos los sistemas operativos, este tendrá sus ventajas e inconvenientes. Aquí veremos algunas de sus ventajas:

- Al estar diseñado para unos dispositivos en concreto, la optimización de código es máxima.
- La gran mayoría de los dispositivos durante su ciclo de vida se mantienen completamente actualizados.
- Seguridad y exigencia en su proceso de publicación de aplicaciones a través de la App Store.

Algunas de sus desventajas:

- Utiliza lenguajes de programación menos conocidos.
- Necesidad de una cuenta con suscripción anual de pago para su despliegue.
- Poca libertad en cuanto a la personalización de aplicaciones.

2.1.3 Windows

Si se desea un máximo despliegue hacia la mayor cantidad de equipos Windows, la opción más interesante es trabajar con una versión concreta y limitada de Windows como es la nueva plataforma universal de Windows (UWP, Universal Windows Platform), siendo una plataforma compatible con la totalidad de dispositivos que ejecutan Windows 10 sin importar si se trata de equipos más o menos complejos, desde un equipo de sobremesa hasta una Surface. Permitiendo que un único paquete de instalación sea compatible con la totalidad de los equipos que utilicen este sistema. Como se ve en la siguiente ilustración, donde tenemos la totalidad de familias compatibles con este nuevo sistema de aplicaciones.



Ilustración 6: Familia de dispositivos Windows UWP

Es posible que existan problemas de compatibilidad con algunos paquetes que si estarían disponibles para una aplicación común de escritorio, pero gracias a una gran comunidad tras esta plataforma es posible instalar *plugins* compatibles que realicen la misma tarea de forma transparente para el usuario (Microsoft, 2018).

Algunas de sus ventajas son las siguientes:

- Instalación más rápida y sencilla.
- Utilizan un centro de actualizaciones integrado para estas Apps.
- Aprovechan todas las funcionalidades que proporcione el equipo en el que se ejecutan.
- Gracias a su funcionamiento interno, la seguridad es mayor que en las aplicaciones comunes de escritorio.

También tienen sus desventajas:

- Falta de despliegue de Windows 10.
- No existe retro compatibilidad con versiones anteriores del sistema operativo, es posible acceder a una aplicación UWP de forma remota desde sistemas operativos anteriores, pero se resentiría el rendimiento.

2.2 APLICACIONES WEB

También llamadas *web Apps*, las aplicaciones son codificadas utilizando un lenguaje de programación soportado por los navegadores web y a su vez ejecutadas por estos. Basadas en HTML (HyperText Markup Language, lenguaje de marcas de hipertexto) junto a Javascript para realizar un contenido dinámico y CSS (Cascading Style Sheets, hoja de estilos en cascada) de forma que podamos aplicar un estilo personalizado a nuestra aplicación (Cuello et al, 2013).

Son aplicaciones diseñadas para cualquier plataforma, utilizando elementos genéricos que no permiten identificar la experiencia con la de un sistema operativo en concreto, todo lo contrario que sucedía en las aplicaciones nativas. La única e indispensable condición es la de disponer de un navegador web en el dispositivo y la consiguiente conexión a internet. Son aplicaciones que no se encuentran físicamente instaladas, si no que su acceso se realiza a través del navegador. Este tipo de acceso permite que el contenido de estas aplicaciones se encuentra fácilmente actualizado, basta con que el desarrollador actualice



el contenido en el servidor web para que los usuarios se aprovechen de ello (Gómez, 2016).

Al igual que existen múltiples tipos de desarrollo para una aplicación móvil, en este apartado existen variedades de tipos de *web App* en función de la complejidad que se desee desarrollar.

- Aplicación web estática: Muestran poca información, de las 3 tecnologías en las que suelen estar basadas las aplicaciones web solo hacen uso de 2 de estas, HTML y CSS. La única posibilidad que existe para editar su contenido es la de descargar el fichero HTML referente a una página en concreto, para posteriormente volver a subirlo al servidor y de este modo actualizarlo.
- Aplicación web dinámica: Más complejas que las anteriores, a través de una capa de PHP (Hypertext Preprocessor, procesador de hipertexto) o ASP (Active Server Pages, páginas de servicio activo) permiten que el contenido a mostrar sea dinámico, trabajando junto a bases de datos para la actualización de la información.
- Tienda virtual o comercio electrónico: A la complejidad de las aplicaciones dinámicas se le añade la gestión en cuanto a los métodos de pago, siendo más importante si cabe toda la gestión de seguridad del sitio web.
- Portal *web App*: Aplicación la cual agrupa páginas con diferentes tipos de contenido, desde foros a chats o correo electrónico.
- Aplicación web animada: Junto a las tecnologías ya vistas se le añade FLASH. Permite presentar contenidos animados, creativos y modernos. Al no permitir que los buscadores lean correctamente la información de esta tecnología hace que se vea perjudicado su posicionamiento web.
- Aplicaciones web con gestor de contenidos: Hace necesario el instalar un gestor de contenidos (CMS), de este modo el administrador puede realizar y actualizar el sitio web sin acceder directamente a él. Estos gestores, como por ejemplo WordPress, permiten la creación y mantenimiento de sitios web sin hacer necesario tener un conocimiento interno de la aplicación, a través de plantillas y configuraciones predefinidas cualquier usuario puede crear su sitio web de forma rápida, y asociarlo a su dominio para tener acceso a él desde cualquier navegador.

Este tipo de aplicaciones tiene sus ventajas e inconvenientes, es importante remarcarlos a la hora de comprobar si es lo más conveniente para el desarrollo que interese (Cabello, 2014).

Ventajas:

- Es una de las opciones más baratas, se desarrolla una aplicación multiplataforma mediante únicamente conocimientos relacionados con el desarrollo WEB, el desarrollador solo debe preocuparse de su compatibilidad con estos navegadores.
- Para su acceso únicamente será necesaria su dirección URL a través del navegador, haciendo que el despliegue sea extremadamente simple, así como si llegada a los usuarios. No es necesario instalar nada en la parte del cliente.
- La información se encuentra centralizada en un servidor.

Desventajas:

- Debe estar diseñada mediante un diseño responsivo, no basta con que se adapte a la pantalla de un dispositivo concreto, debe adaptarse a las pantallas de la totalidad de los dispositivos a los que se pretenda llegar.
- Necesidad de conexión a internet para su utilización.
- Menor visibilidad que si se encontrara en una tienda de aplicaciones.

- Restricción de acceso a ciertas características Hardware del dispositivo.
- Su tiempo de respuesta puede ser algo más lento debido a la conexión a internet.

2.3 APLICACIONES HÍBRIDAS

Este tipo de aplicaciones son una combinación de los dos tipos anteriores que se han visto. Su desarrollo es similar al de las aplicaciones web. La única diferencia consiste en que una vez finalizada la aplicación web, debe empaquetarse como si de una aplicación nativa se tratase. Mediante un mismo código desarrollado para navegador es posible obtener una aplicación nativa y única por cada plataforma y de este modo realizar una distribución a través de las diferentes tiendas de aplicaciones (Cuello et al, 2013).

Para el usuario final da la impresión de un trabajo más logrado, bien es cierto que sigue sin utilizar los elementos característicos del sistema operativo en el que se utiliza, pero permite trabajar con algunas de las capacidades internas que en el caso de trabajar desde un navegador web no sería posible. Este acceso a las capacidades internas marcará la diferencia entre los sistemas operativos a los que llegará, ya que el código utilizado en cada plataforma será único, y será el desarrollo a mayores que necesitará comparado con una *web App*. Sin olvidar la mejoría en cuanto a rendimiento si se compara trabajar a través de un navegador con trabajar desde la propia aplicación, siempre que la complejidad no sea excesiva.

En términos de mantenimiento tiene las ventajas de una aplicación web, salvo en lo referente al acceso a las capacidades internas del dispositivo, siendo únicamente necesario mantener un único código para todas las plataformas añadiendo el trabajo de actualizar la aplicación en las respectivas tiendas.

Entrando con detalle en su desarrollo existen diferentes frameworks que permiten el entrar de lleno en la producción de aplicaciones híbridas. PhoneGap y Apache Cordova son herramientas muy similares que permiten este tipo de desarrollos.



Ilustración 7: Logos PhoneGap y Apache Cordova

Ambas opciones son gratuitas, de código abierto y se utilizan del mismo modo. PhoneGap, propiedad de Adobe es una distribución de Apache Cordova. Con lo que los servicios que ofrecen son similares entre ambas herramientas. El único añadido es el servicio de compilación para PhoneGap que proporciona Adobe, haciendo más simple su trabajo sin ser necesaria la instalación del SDK para su compilación. En el caso de proyectos de código abierto este servicio es gratuito, todo lo contrario que si se trabaja en una aplicación comercial.

Centrándonos exclusivamente en las ventajas e inconveniente de una aplicación híbrida tenemos las siguientes (Ortuño, 2016).

Ventajas:

- Acceso a la gran mayoría de las capacidades del dispositivo.
- Mantenimiento único para el código compatible a todas las plataformas.
- Conectividad online y offline.
- Menor coste de desarrollo en aplicaciones multiplataforma.

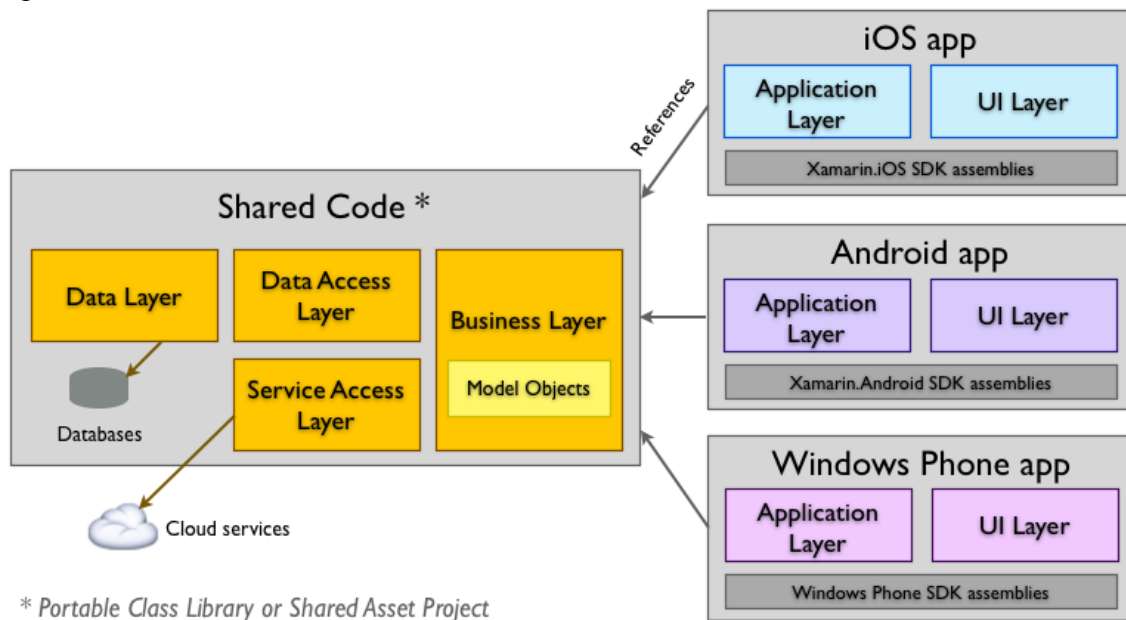
Desventajas:

- A pesar de tener un buen rendimiento comparado con una aplicación web, nunca llegaremos al rendimiento de las aplicaciones nativas. En el caso de que la aplicación sea una aplicación muy compleja ello afectará notablemente al rendimiento.
- Las tiendas de aplicaciones ponen bastantes restricciones al desarrollo de aplicaciones híbridas, siendo beneficiosas de cara a las aplicaciones nativas.

2.4 APLICACIONES GENERADAS

Se trata de una tecnología relativamente nueva, aun no muy extendida con algunas compañías apostando fuerte por este tipo de desarrollo. Sus aplicaciones son desarrolladas con un lenguaje de programación específico del software utilizado, una vez desarrollado y a través de este código, se genera un código nativo independiente para cada plataforma a desarrollar. Comparten con las híbridas el hecho de reciclar código, a pesar de ello, no se parecen en nada a las aplicaciones híbridas, ya que, en el caso de las aplicaciones generadas, este código estándar se adapta a cada sistema operativo concreto.

El comportamiento y desarrollo de las aplicaciones nativas sería el que se ve en la siguiente ilustración:



* Portable Class Library or Shared Asset Project

Ilustración 8: Desarrollo software aplicaciones generadas

A través de un código compartido el compilador es el encargado de generar el código nativo para Android, iOS o Windows.



A pesar de trabajar con un código independiente al de cada plataforma en concreto, es importante el conocer algo de cada lenguaje específico. El hecho de desarrollar una aplicación multiplataforma supone un coste claramente menor al de las aplicaciones nativas, haciendo que el rendimiento sea prácticamente similar, lo que puede hacer muy rentable decidir utilizar esta tecnología antes que aplicaciones web o híbridas, con un pequeño coste a mayores el rendimiento es claramente superior.

Para la parte de pruebas, emulación y simulación de las aplicaciones generadas el software existente es extremadamente exigente de momento. Para iOS será indispensable el contar con un equipo macOS en la misma red que el equipo de desarrollo, a fin de poder compilar y probar a través de Xcode la aplicación o bien en uno de sus emuladores de iPhone o iPad, o bien en un dispositivo físico conectado por cable al equipo macOS. Para Windows debe ser compilado en un PC con Windows 10 y este mismo equipo trae consigo los diferentes simuladores de dispositivos necesarios. El caso de Android es el más especial basta con tener instalado el SDK correspondiente de Android y el mismo software trae consigo los emuladores necesarios para hacer pruebas, o si se desea teniendo el Smartphone o Tablet en modo depuración, es posible compilar y ejecutar paso a paso las aplicaciones directamente en el dispositivo.

2.5 SOLUCIÓN ESCOGIDA Y JUSTIFICACIÓN

Tras un recorrido a través de todas las posibles tecnologías para el desarrollo de la aplicación ahora es el momento de la decisión más importante de todas, la elección del software correcto que permita el desarrollo de esta aplicación para el tratamiento de pacientes en cuanto a la evaluación y rehabilitación de las capacidades cognitivas. Para ello es importante ir haciéndose una serie de preguntas a fin cerrar el cerco lo más posible hacia la tecnología más conveniente.

No será necesaria la decisión de sobre qué sistema operativo trabajar, se pretende la creación de una aplicación multiplataforma capaz de trabajar en cualquiera de los tres sistemas operativos móviles con mayor penetración del mercado sin importar cuál de los tres se utilice. Es por ello que la decisión a tomar es la tecnología y el software a utilizar para el desarrollo de esta.

Hay una tecnología que se puede descartar sin entrar en muchos detalles, es imposible el realizar esta aplicación como una aplicación Web, ni por rendimiento, ni por método de uso. El objetivo es el de poder trabajar tanto online como offline, cosa imposible si es necesaria una conexión a internet activa en todo momento. A partir de este punto ya solo debemos de tener en cuenta los tres tipos restantes.

El tiempo que tardemos en desarrollar la aplicación es un factor más que importante, el tiempo es dinero, y si en este proceso el tiempo es excesivo se pierde este recurso. No es posible invertir el tiempo necesario en el desarrollo de tres aplicaciones nativas independientes, en el caso de un mismo encargado trabajando con las tres plataformas, el tiempo de desarrollo será excesivo ya que requiere un aprendizaje de cada plataforma con su posterior programación y pruebas en cada una de ellas de forma independiente. Y en el caso de un encargado para el desarrollo de cada plataforma, se necesitaría el triple de personal, con su consiguiente gasto. El tiempo que se invertiría en la aplicación sería menor, pero debería de tenerse un consenso de modo que el funcionamiento, apariencia y rendimiento utilizándose cualquiera de las tres plataformas fuera el mismo. Es por ello



que perdemos otra de las posibles opciones, no podemos desarrollar esta aplicación como tres aplicaciones nativas independientes.

Nos quedan dos posibles opciones, una aplicación híbrida o una aplicación generada. Ambas con algunas similitudes y algunas diferencias. A pesar de tener en ambas un código común, las aplicaciones generadas requieren de algo más de conocimiento en cuanto a cada plataforma de manera independiente. Esto también afecta a su rendimiento, haciendo que su rendimiento sea muy próximo al de una aplicación nativa. Ahora se debe pensar, ¿merece la pena ese coste mayor en materia de investigación en favor de una mejora significativa de rendimiento? La respuesta claramente es sí.

La mejor opción disponible para el tipo de desarrollo que se necesita es la de desarrollar una aplicación generada. Es cierto que requiere una curva de aprendizaje mayor, no solo basta con aprender ese código común que utilice el software elegido, sino que se debe hacer especial hincapié en todos esos puntos donde será necesario acceder a características concretas de cada dispositivo, haciendo uso de código concreto e independiente para cada plataforma. Aquí es donde buscamos el equilibrio entre coste y rendimiento, no llegaremos al rendimiento perfecto que daría como resultado una aplicación nativa al 100%, pero sí que nos aproximamos enormemente a este con un coste mucho menor.

En el caso de la solución elegida ahora se debe encontrar el framework adecuado que posibilite el desarrollo de la aplicación. Es una tecnología especialmente novedosa, y pocas son las compañías que están apostando por ella en un principio, lo que provoca un menor mercado de herramientas donde comenzar a desarrollar. Bien es cierto que no ha sido necesario buscar mucho hasta dar con la herramienta idónea. Microsoft ha tenido problemas en los últimos años con las aplicaciones que ha ido diseñando, su cuota de mercado cada vez es menor en dispositivos móviles en cambio la de sus máximos competidores no hace más que crecer. Los últimos sistemas operativos tanto para ordenador como para Smartphone diseñados por Microsoft no han triunfado como se esperaba, el gran fracaso de Windows Vista y el poco despliegue de Windows 8 o de Windows Phone llevaron a Microsoft a crear una nueva familia de dispositivos con un sistema operativo común para todos como fue Windows 10. Y con ello han dado facilidades a desarrolladores de otras plataformas para crear aplicaciones utilizando su propio software.

Como herramienta de trabajo se ha decidido utilizar Visual Studio 2017, mediante la API (Application Programming Interfaces, Interfaces de programación de aplicaciones) de Xamarin. A continuación, se presenta los motivos por los que se ha escogido esta herramienta:

- Utiliza un lenguaje fácil de manejar como es C#.
- Posibilidad de crear soluciones multiplataforma y desplegarlas rápidamente.
- La gran comunidad de Microsoft que hay detrás permite encontrar una amplia cantidad de manuales de ayuda para su desarrollo.
- Software libre para su desarrollo multiplataforma, el software Xamarin se incluye de forma gratuita con la versión Community de Visual Studio.

Para proseguir se va a entrar con más detalle en todas y cada una las características de Xamarin.





3. XAMARIN

Xamarin es un entorno de desarrollo, el cual, permite crear aplicaciones multiplataforma para Android, iOS y Windows utilizando un lenguaje de programación común, en este caso C#. No es necesario aprender el lenguaje de una plataforma específica, es decir, facilita la vida al programador ya que este puede trabajar únicamente conociendo las características propias de xamarin y de este lenguaje de programación. Existen varios frameworks con Xamarin integrado, sin embargo, el más recomendado es visual studio, propiedad de Microsoft.

Xamarin permite combinar el rendimiento logrado en aplicaciones nativas con unas características eficaces (Xamarin Microsoft, 2017):

- Xamarin contiene enlaces para casi todos los SDK de las plataformas de iOS y Android, siendo compatible con diferentes versiones produciendo menos errores en tiempo de ejecución.
- Interoperabilidad con bibliotecas específicas de cada plataforma, permitiendo invocar las bibliotecas de Objective-C, Java, C y C++. Permite utilizar código de terceros en nuestras aplicaciones para el código de una plataforma en concreto.
- Compatibilidad multiplataforma móvil: Ofrece compatibilidad con las tres principales plataformas móviles, con una API unificada que permite acceder a los recursos de las tres plataformas. permitiendo reducir el tiempo de desarrollo de las aplicaciones.

Incluidas en Xamarin existen diferentes herramientas para facilitar tanto el desarrollo como el diseño de la aplicación, la más conocida de ellas es Xamarin.Forms.

3.1 XAMARIN.FORMS

Se trata de un kit de herramientas de interfaz de usuario multiplataforma, permitiendo crear diseños semejantes a los de las aplicaciones nativas, con una diferencia, los elementos se comparten entre las aplicaciones de todas las plataformas cambiando según en la que se ejecute. Existen unas limitaciones en cuanto al sistema operativo de la plataforma destino (Xamarin Forms, 2016):

- iOS 8 o versiones posteriores.
- Android 4.0.3 o versiones posteriores.
- Windows: Plataforma universal de Windows 10, Windows UWP.

Estas aplicaciones se pueden desarrollar tanto en equipos Windows como macOS, sin embargo es obligatorio su uso en Windows junto con Visual Studio para el desarrollo de su versión en Windows, si junto a estas versiones se desea poder compilar y ejecutar en iOS será necesario disponer de un equipo MAC en la misma red.

Al poderse considerar aplicaciones nativas las generadas mediante Xamarin.Forms, no tienen las limitaciones de otras aplicaciones multiplataforma en cuanto al espacio aislado, el acceso limitado a APIs o su bajo rendimiento. Estas aplicaciones pueden utilizar cualquier API así, acceder a cualquier recurso del dispositivo. No es necesario crear toda la interfaz gráfica de la aplicación mediante Xamarin.Forms, es posible crear partes mediante el kit de herramientas nativo y diseñar esta combinando ambas.

Tenemos dos técnicas para la creación de interfaces gráficas al igual que sucede con las aplicaciones nativas:

- Mediante código fuente C#: La interfaz gráfica se genera junto al código ejecutable, como sucede por ejemplo en Android la creación de interfaz gráfica a través de código es menos recomendable, supone un mayor consumo para el procesador.

```
var layout = new StackLayout();
var button = new Button
{
    Text = "StackLayout",
    VerticalOptions = LayoutOptions.Start,
    HorizontalOptions = LayoutOptions.FillAndExpand
};
var yellowBox = new BoxView { Color = Color.Yellow, VerticalOptions = LayoutOptions.FillAndExpand,
    HorizontalOptions = LayoutOptions.FillAndExpand };
var greenBox = new BoxView { Color = Color.Green, VerticalOptions = LayoutOptions.FillAndExpand,
    HorizontalOptions = LayoutOptions.FillAndExpand };
var blueBox = new BoxView
{
    Color = Color.Blue,
    VerticalOptions = LayoutOptions.FillAndExpand,
    HorizontalOptions = LayoutOptions.FillAndExpand,
    HeightRequest = 75
};

layout.Children.Add(button);
layout.Children.Add(yellowBox);
layout.Children.Add(greenBox);
layout.Children.Add(blueBox);
layout.Spacing = 10;
Content = layout;
```

Ilustración 9: Ejemplo de diseño interfaz C#

- Uso de XAML (eXtensible Application Markup Language, Lenguaje Extensible de Formato para Aplicaciones): Lenguaje especialmente diseñado para la creación de la interfaz de usuario base en la presentación de Windows. Se basa en XML (eXtensible Markup Language, Lenguaje de Marcado Extensible) optimizado para su uso en interfaces gráficas complejas haciéndolo más sencillo e intuitivo. La mayor ventaja es que el diseño a través de XAML no requiere de ningún tipo de programación siendo relativamente sencillo su diseño si se conoce ligeramente la estructura de etiquetas y propiedades.


```
<?xml version="1.0" encoding="UTF-8"?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
x:Class="LayoutSamples.StackLayoutDemo"
Title="StackLayout Demo">
  <ContentPage.Content>
    <StackLayout Spacing="10" x:Name="layout">
      <Button Text="StackLayout" VerticalOptions="Start"
HorizontalOptions="FillAndExpand" />
      <BoxView Color="Yellow" VerticalOptions="FillAndExpand"
HorizontalOptions="FillAndExpand" />
      <BoxView Color="Green" VerticalOptions="FillAndExpand"
HorizontalOptions="FillAndExpand" />
      <BoxView HeightRequest="75" Color="Blue" VerticalOptions="End"
HorizontalOptions="FillAndExpand" />
    </StackLayout>
  </ContentPage.Content>
</ContentPage>
```

Ilustración 10: Ejemplo interfaz mediante XAML

De este modo es posible generar una misma interfaz gráfica de dos métodos completamente diferentes, y como vemos en la ilustración el resultado es el mismo con la diferencia de la apariencia de los elementos como si de elementos nativos se tratase.

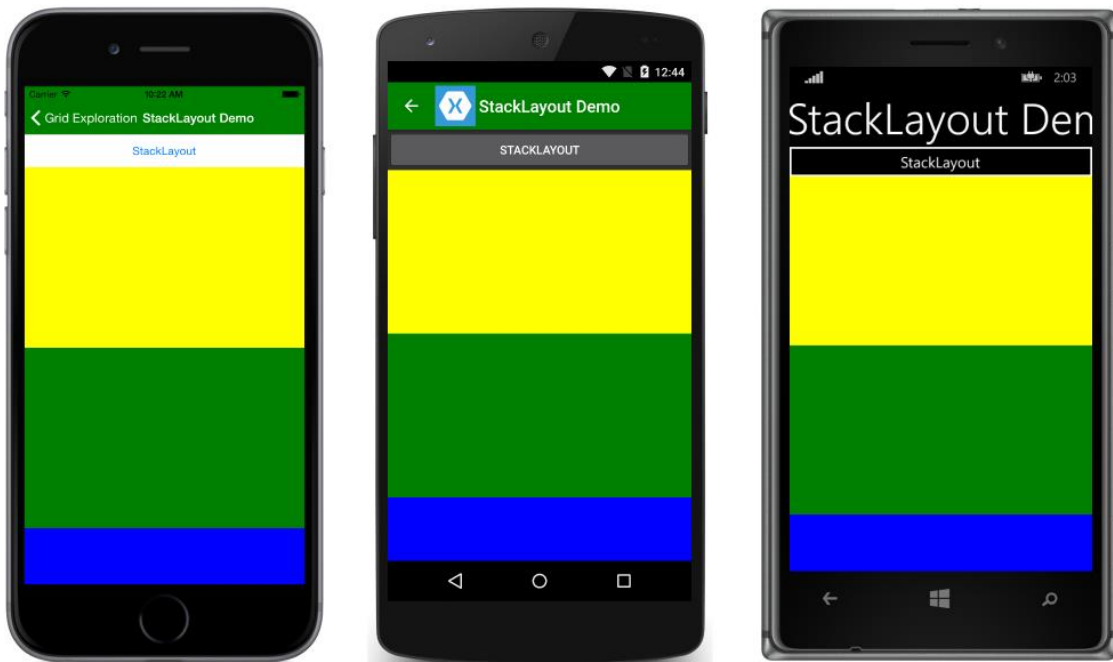


Ilustración 11: Interfaz gráfica resultante con ambos códigos

En cuanto a la arquitectura de Xamarin.Forms se basa en la maximización del código a compartir entre plataformas. Respetando los principios de programación orientada a objetos para obtener una aplicación con capas diferenciadas e independientes:

- Encapsulación: garantizar que las capas y clases llevan a cabo sus funciones ocultando los detalles de su implementación. No es necesario saber cómo realizan las tareas de forma interna los objetos.

- Separación de responsabilidades: cada componente debe realizar únicamente sus funciones y exponer sus funcionalidades a través de una API accesible para los demás.
- Polimorfismo: programación a una interfaz compartida entre plataformas la cual interactúa con características específicas de la plataforma.

A nivel de arquitectura existen dos tipos de proyectos en función de la librería de clases que se utilice, podemos estar ante un proyecto compartido (Shared Project) o bien en un proyecto con una PCL (Portable Class Library, Librería de clases portátil).

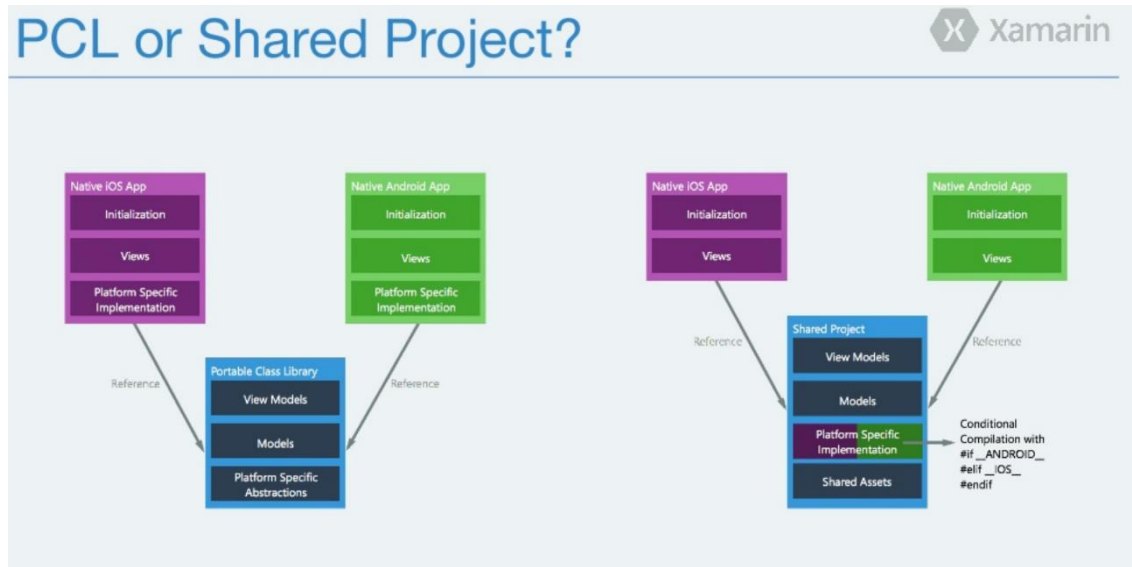


Ilustración 12: Arquitectura proyectos PCL y Shared Projects

- PCL: Para la generación de aplicaciones multiplataforma independientes, es necesario, un proyecto por cada plataforma. A estos tres proyectos se les añade un proyecto PCL compatible con ellas. El código del proyecto PCL se compila contra sus propias referencias, estas librerías son compatibles con Android, iOS y Windows. Creando un fichero .dll para cada plataforma y finalmente pudiendo crear un fichero .APK, .IPA o .APP según el sistema operativo destino.

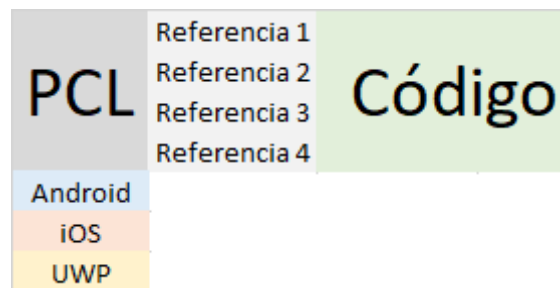


Ilustración 13: Ejemplo proyecto PCL

- Shared Projects: El proyecto compartido no produce ningún .dll, es una agrupación del código que se compila directamente contra las propias referencias de los proyectos independientes de cada sistema operativo destino. Debemos tener

referenciadas las mismas librerías en cada proyecto independiente, de no ser así, la compilación devolvería un error en una de las tres plataformas con la notificación de no conocer la referencia que se le pide.

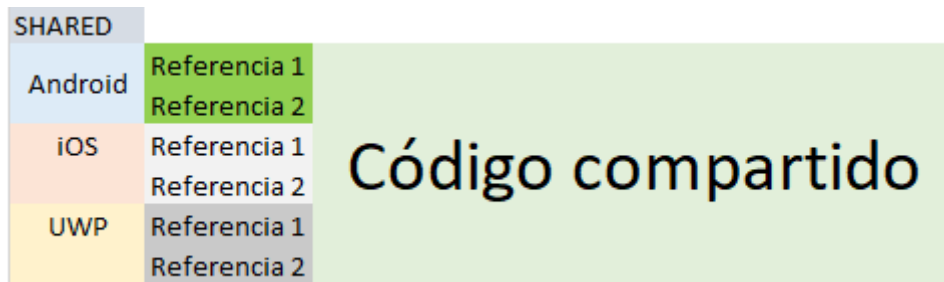


Ilustración 14: Ejemplo Shared Projects

Al utilizar librerías separadas para cada solución, es posible que no todas tengan un código compatible entre ellas, aquí es donde viene la compilación condicional. La cual nos permite distinguir secciones de código de modo que únicamente se compilen contra las librerías de un sistema operativo concreto, siendo invisibles para los demás.

```
#if __ANDROID__
    Android.Media.MediaPlayer player = new Android.Media.MediaPlayer();
#elif WINDOWS_UWP
    Windows.Media.Playback.MediaPlayer player = new Windows.Media.Playback.MediaPlayer();
#elif __IOS__
    AVFoundation.AVAudioPlayer player;
#endif
```

Ilustración 15: Ejemplo de compilación condicional

Si se entra en el contexto de la aplicación a desarrollar, se optó por la opción de un proyecto compartido (Shared Project). Para el acceso a las capacidades internas del dispositivo en algunas ocasiones es necesario de utilizar librerías de terceros. Esto provoca que pueda haber problemas de compatibilidad, como así fue. Recurriendo a diferentes bibliotecas para cada solución, siendo necesario utilizar una compilación condicional siempre que hubiera que acceder a estas.





4. ANÁLISIS

4.1 ACTORES DEL SISTEMA

Se distinguen tres tipos de actores con la posibilidad de interactuar con el sistema: Paciente, Terapeuta y la propia aplicación EntrenaTu.

ACT – 01	Paciente
Versión	1.0 (11/06/2018)
Autor	Marcos Muñoz Morales
Descripción	Este actor representa a cualquiera de los pacientes de la clínica con acceso a la aplicación y tratamiento activo disponible
Comentarios	Debe conocer su imagen de clave asignada para poder iniciar el tratamiento.

ACT – 02	Terapeuta
Versión	1.0 (11/06/2018)
Autor	Marcos Muñoz Morales
Descripción	Este actor representa a los diferentes terapeutas de la clínica con diferentes pacientes asignados, pudiendo modificar el paciente asignado a la aplicación en cada dispositivo móvil. Los cuales controlan la evaluación de sus pacientes en función del resultado de los tratamientos realizados.
Comentarios	Se identifica a través su usuario y clave contra el servidor.

ACT – 03	EntrenaTu
Versión	1.0 (11/06/2018)
Autor	Marcos Muñoz Morales
Descripción	Este actor representa a la propia aplicación, la cual, realiza algunas tareas transparentes al resto de actores en tiempo de ejecución. Su función más importante es la de mantener actualizada la información del tratamiento del paciente en la aplicación, así como los resultados del servidor.
Comentarios	La aplicación debe ser ejecutada por uno de los actores anteriores para que se realicen sus tareas

4.2 DIAGRAMA DE CASOS DE USO

En este apartado se puede ver un diagrama de casos de uso. En este se muestran los actores definidos en el apartado anterior caracterizando las acciones que pueden realizar cada uno de ellos.

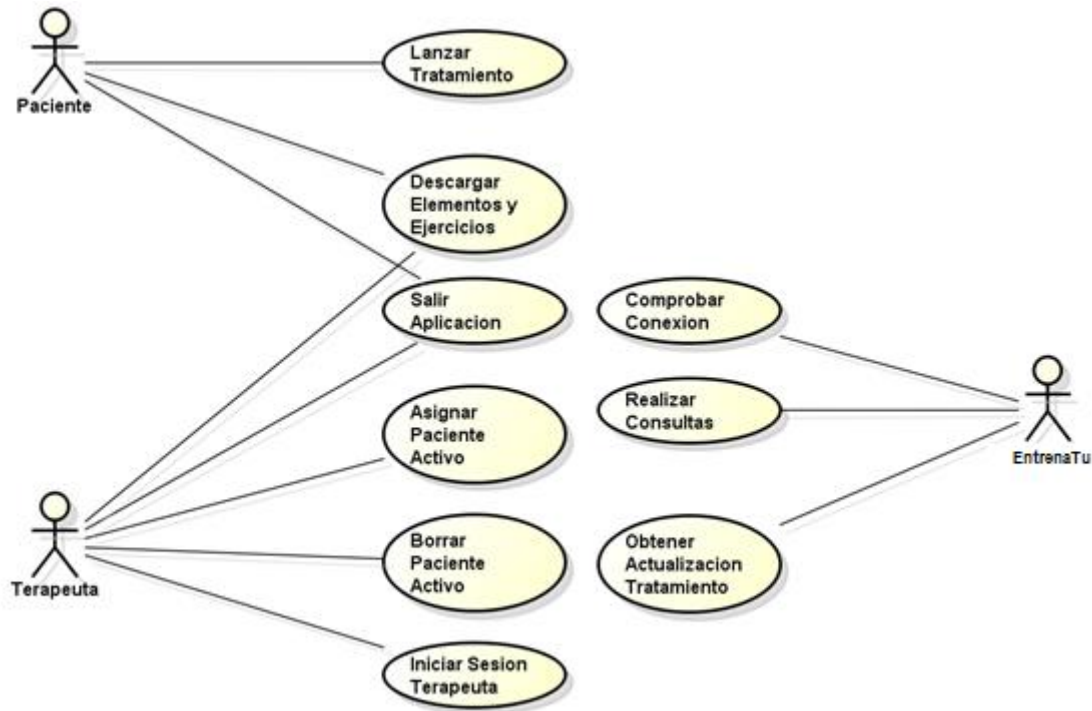


Ilustración 16: Diagrama de Casos de Uso

4.3 ESPECIFICACIONES DE LOS CASOS DE USO

En este apartado se va a especificar cada uno de los casos de uso que han sido plasmados en el diagrama anterior.

4.3.1 Lanzar tratamientos

Identificador	Lanzar Tratamientos	
Versión	1.0	
Fecha de la última revisión	11/06/2018	
Autores	Marcos Muñoz Morales	
Descripción	El paciente realiza el tratamiento asignado por el terapeuta	
Personal involucrado o Actores	Paciente	
Precondición	El terapeuta debe haber asignado previamente como paciente activo en el dispositivo a este paciente. Se deben haber descargado los elementos y ejercicios antes de poder iniciar cualquier tratamiento.	
Escenario principal de éxito o Secuencia o Flujo Normal	Paso	Escenario principal de éxito o Secuencia o Flujo Normal
	1	El paciente hace clic sobre su foto o nombre.



	2	El sistema muestra las posibles claves para iniciar el tratamiento
	3	El Paciente hace clic sobre su clave personal.
	4	El sistema carga el siguiente ejercicio asignado al paciente
	5	El usuario realiza los ejercicios pertenecientes a la sesión correspondiente
	6	El sistema almacena los resultados y muestra las estadísticas de la sesión al paciente
Post condición	Se ha completado el tratamiento con éxito.	
Alternativa o Flujo Alternativo	Paso	Alternativa o Flujo Alternativo
	1-5	El paciente puede cerrar la aplicación en cualquier momento, el caso de uso finaliza sin llegar a la post condición.
	1-3	Si el paciente hace clic en salir el caso de uso finaliza sin llegar a la post condición.
Requisitos Especiales	Ninguno	
Frecuencia esperada	Alta	
Importancia	Alta	
Urgencia	PD	
Comentarios	Ninguno	

Tabla 1: Especificaciones caso de uso lanzar tratamiento

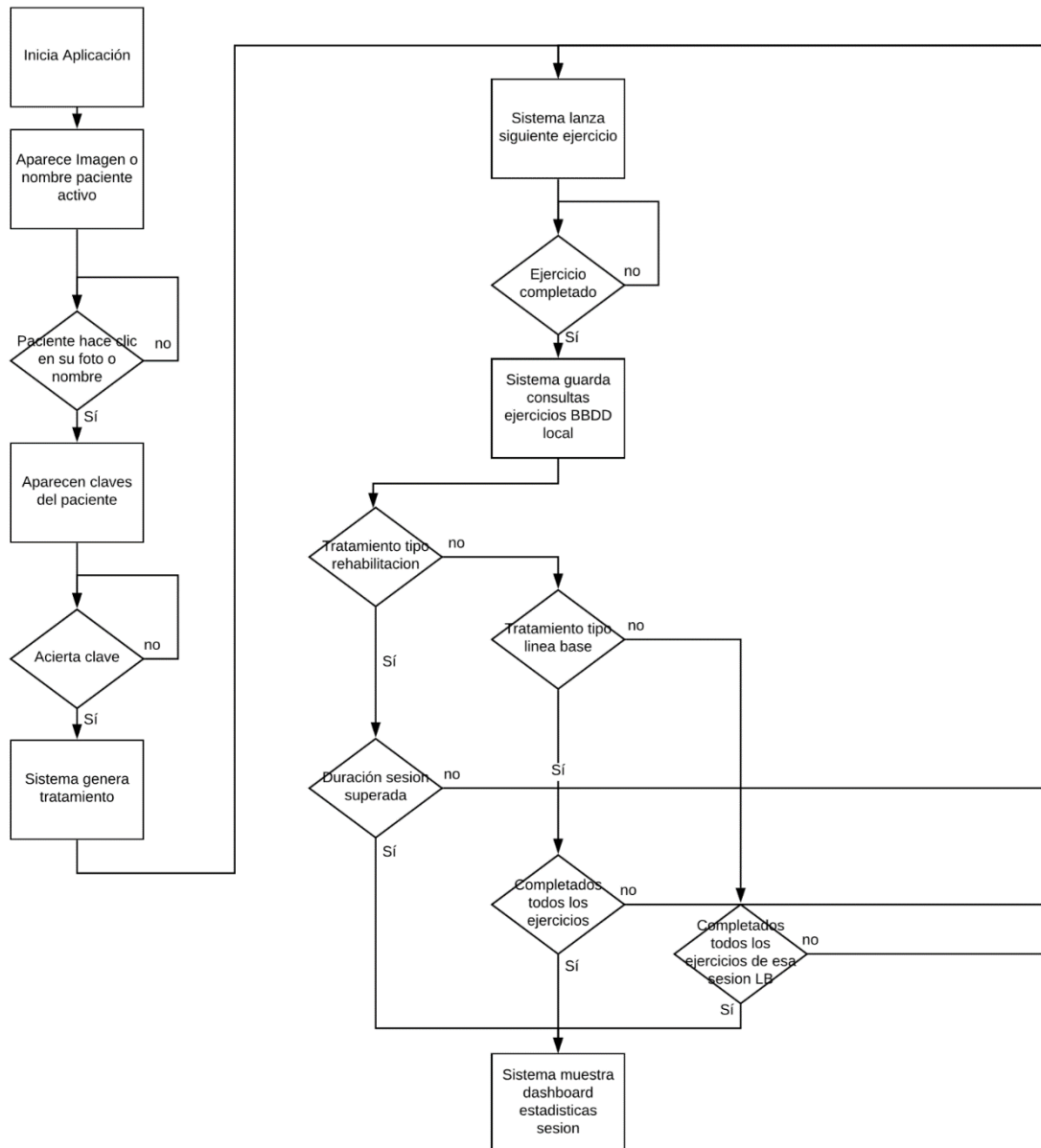


Ilustración 17: Diagrama de flujo lanzar tratamiento

4.3.2 Descargar elementos y ejercicios

Identificador	Descargar Elementos y Ejercicios
Versión	1.0
Fecha de la última revisión	11/06/2018
Autores	Marcos Muñoz Morales
Descripción	En caso de no tener almacenados los elementos (imágenes, textos y audios) y ejercicios en la base de datos local de SQLite, se solicita al paciente o terapeuta si desea descárgalos.
Personal involucrado o Actores	Paciente y Terapeuta



Precondición	No deben haberse descargado previamente todos los elementos o ejercicios. Se debe tener conexión con el servidor.	
Escenario principal de éxito o Secuencia o Flujo Normal	Paso	Escenario principal de éxito o Secuencia o Flujo Normal
	1	El paciente o terapeuta hace clic en el botón de descargar elementos.
	2	El sistema descarga elemento a elemento hasta completar la descarga de elementos.
	3	El paciente o terapeuta puede ver el proceso de descarga de elementos.
	4	El sistema descarga ejercicio a ejercicio hasta completar la descarga de ejercicios.
	5	El paciente o terapeuta puede ver el proceso de descarga de ejercicios.
	6	El sistema pasa a la pantalla de acceso de paciente o terapeuta informando de la descarga completada.
Post condición	Los elementos y ejercicios se han guardado en la BBDD local de la aplicación en el dispositivo.	
Alternativa o Flujo Alternativo	Paso	Alternativa o Flujo Alternativo
	2	En caso de que el sistema detecte que ya se contienen todos los elementos se salta al paso 4.
	4	En caso de que el sistema detecte que ya se contienen todos los ejercicios se salta al paso 6.
Requisitos Especiales	Ninguno	
Frecuencia esperada	Baja	
Importancia	Alta	
Urgencia	PD	
Comentarios	Ninguno	

Tabla 2: Especificaciones caso de uso descargar elementos y ejercicios

4.3.3 Salir de la aplicación

Identificador	Salir de la aplicación
Versión	1.0
Fecha de la última revisión	11/06/2018
Autores	Marcos Muñoz Morales
Descripción	Tanto terapeuta como paciente pueden cerrar la aplicación estando en los menús.



Personal involucrado o Actores	Paciente y Terapeuta	
Precondición	Se debe haber iniciado la aplicación. No haber iniciado ninguna sesión de tratamiento o en su defecto haberla terminado.	
Escenario principal de éxito o Secuencia o Flujo Normal	Paso	Escenario principal de éxito o Secuencia o Flujo Normal
	1	El paciente o terapeuta hace clic en el botón “Salir”.
	2	El sistema cierra la aplicación.
Post condición	La aplicación se cierra con éxito	
Requisitos Especiales	Ninguno	
Frecuencia esperada	Media	
Importancia	Media	
Urgencia	PD	
Comentarios	Al ser una aplicación móvil se puede cerrar en cualquier momento, pero es recomendable cerrarla a través del botón creado para ello y en los momentos en los que está disponible..	

Tabla 3: Especificaciones caso de uso salir de la aplicación

4.3.4 Asignar paciente activo

Identificador	Asignar paciente activo	
Versión	1.0	
Fecha de la última revisión	11/06/2018	
Autores	Marcos Muñoz Morales	
Descripción	El terapeuta asignará un paciente activo al dispositivo	
Personal involucrado o Actores	Terapeuta	
Precondición	Se debe tener conexión a internet y que el paciente que se desea asignar tenga un tratamiento activo en la base de datos del servidor.	
Escenario principal de éxito o Secuencia o Flujo Normal	Paso	Escenario principal de éxito o Secuencia o Flujo Normal
	1	El terapeuta hace clic en el botón de acceso como terapeuta.
	2	El sistema muestra la pantalla para que el terapeuta se autentique.



	3	El terapeuta introduce su usuario y contraseña para después hacer clic en acceder.
	4	El sistema comprueba que coinciden usuario y contraseña y carga la pantalla de selección de paciente activo.
	5	El terapeuta selecciona de la lista el paciente que desea asignar como activo al dispositivo y hace clic en asignar y volver.
	6	El sistema muestra un mensaje a cerca de usuario asignado con éxito y vuelve a la pantalla inicial.
Post condición	El paciente asignado podrá ejecutar su tratamiento en futuras ejecuciones de la aplicación.	
Alternativa o Flujo Alternativo	Paso	Alternativa o Flujo Alternativo
	4	Si el usuario y contraseña no coinciden el sistema informa de ello, volviendo al punto 3 para un nuevo intento.
	1-6	Si el terapeuta hace clic en el botón salir en cualquier momento el caso de uso finaliza sin lograr la post condición.
	1-6	Si el terapeuta hace clic en el botón volver en cualquier momento vuelve a la pantalla de inicio de la aplicación sin lograr la post condición.
	5	Si no hay pacientes activos con tratamiento en la base de datos del servidor el terapeuta tendrá una lista vacía, sin poder lograr la post condición.
Requisitos Especiales	Ninguno	
Frecuencia esperada	Media	
Importancia	Media	
Urgencia	PD	
Comentarios		

Tabla 4: Especificaciones caso de uso asignar paciente activo

4.3.5 Borrar paciente activo

Identificador	Borrar paciente activo
Versión	1.0
Fecha de la última revisión	11/06/2018
Autores	Marcos Muñoz Morales
Descripción	El terapeuta borrará el paciente activo del dispositivo.
Personal involucrado o Actores	Terapeuta
Precondición	Se debe tener conexión a internet y debe haber un paciente activo en el dispositivo.



Escenario principal de éxito o Secuencia o Flujo Normal	Paso	Escenario principal de éxito o Secuencia o Flujo Normal
	1	El terapeuta hace clic en el botón de acceso como terapeuta.
	2	El sistema muestra la pantalla para que el terapeuta se autentique.
	3	El terapeuta introduce su usuario y contraseña para después hacer clic en acceder.
	4	El sistema comprueba que coinciden usuario y contraseña y carga la pantalla de selección de paciente activo.
	5	El terapeuta hace clic en el botón borrar paciente activo.
	6	El sistema muestra un mensaje a cerca de usuario borrado con éxito.
Post condición	Deja de haber paciente activo en el dispositivo, siendo necesario asignar uno nuevo para poder ejecutar sesiones.	
Alternativa o Flujo Alternativo	Paso	Alternativa o Flujo Alternativo
	4	Si el usuario y contraseña no coinciden el sistema informa de ello, volviendo al punto 3 para un nuevo intento.
	1-6	Si el terapeuta hace clic en el botón salir en cualquier momento el caso de uso finaliza sin lograr la post condición.
	1-6	Si el terapeuta hace clic en el botón volver en cualquier momento vuelve a la pantalla de inicio de la aplicación sin lograr la post condición.
	5	Si no hay pacientes activos con tratamiento en la base de datos del servidor el terapeuta tendrá una lista vacía, sin poder lograr la post condición.
Requisitos Especiales	Ninguno	
Frecuencia esperada	Media	
Importancia	Media	
Urgencia	PD	
Comentarios		

Tabla 5: Especificaciones caso de uso borrar paciente activo

4.3.6 Iniciar sesión terapeuta

Identificador	Iniciar sesión terapeuta
Versión	1.0
Fecha de la última revisión	11/06/2018



Autores	Marcos Muñoz Morales	
Descripción	El terapeuta accede a la sección para la asignación de paciente activo en el dispositivo.	
Personal involucrado o Actores	Terapeuta	
Precondición	Se debe tener conexión a internet.	
Escenario principal de éxito o Secuencia o Flujo Normal	Paso	Escenario principal de éxito o Secuencia o Flujo Normal
	1	El terapeuta hace clic en el botón de acceso como terapeuta.
	2	El sistema muestra la pantalla para que el terapeuta se autentique.
	3	El terapeuta introduce su usuario y contraseña para después hacer clic en acceder.
	4	El sistema comprueba que coinciden usuario y contraseña y carga la pantalla de selección de paciente activo.
Post condición	El terapeuta ha accedido a la pantalla de administración del paciente activo en el dispositivo.	
Alternativa o Flujo Alternativo	Paso	Alternativa o Flujo Alternativo
	4	Si el usuario o contraseña introducida no son correctas el sistema informa de ello y se vuelve al paso 3 para un nuevo intento de acceso.
	1-3	Si el terapeuta hace clic en volver se finaliza el caso de uso sin llegar a la post condición volviendo a la pantalla de inicio.
	1-3	Si el terapeuta hace clic en salir en cualquier momento se cierra la aplicación sin llegar a la post condición.
Requisitos Especiales	Ninguno	
Frecuencia esperada	Media	
Importancia	Media	
Urgencia	PD	
Comentarios		

Tabla 6: Especificaciones caso de uso iniciar sesión terapeuta

4.3.7 Comprobar conexión

Identificador	Comprobar conexión
Versión	1.0
Fecha de la última revisión	11/06/2018
Autores	Marcos Muñoz Morales



Descripción	La aplicación comprueba por sí misma la conexión con el servidor.	
Personal involucrado o Actores	EntrenaTu	
Precondición	Al iniciar la aplicación	
Escenario principal de éxito o Secuencia o Flujo Normal	Paso	Escenario principal de éxito o Secuencia o Flujo Normal
	1	La aplicación comprueba que está activado alguno de los métodos de conexión a internet disponibles
	2	La aplicación comprueba que tiene conexión con el servidor haciendo un ping a la dirección del servicio utilizado.
	3	El servidor responde con éxito a la prueba de conexión.
	4	La aplicación habilita las conexiones con el servidor.
Post condición	La aplicación permitirá realizar las acciones que necesitan de conexión a internet.	
Alternativa o Flujo Alternativo	Paso	Alternativa o Flujo Alternativo
	1	Si no hay método de conexión a internet activo el caso de uso no llega a realizar la post condición.
	3	Si el servidor no responde con éxito el caso de uso no llega a realizar la post condición y se establece que no hay conexión a internet.
Requisitos Especiales	Ninguno	
Frecuencia esperada	Media	
Importancia	Media	
Urgencia	PD	
Comentarios		

Tabla 7: Especificaciones caso de uso comprobar conexión

4.3.8 Realizar consultas

Identificador	Realizar consultas
Versión	1.0
Fecha de la última revisión	11/06/2018
Autores	Marcos Muñoz Morales
Descripción	La aplicación realiza las consultas de ejercicios y tratamientos ya realizados para almacenar sus resultados en el servidor.
Personal involucrado o Actores	EntrenaTu
Precondición	Debe haber conexión con el servidor.



	Debe haber consultas que realizar bien al iniciar la aplicación o bien al finalizar un tratamiento.	
Escenario principal de éxito o Secuencia o Flujo Normal	Paso	Escenario principal de éxito o Secuencia o Flujo Normal
	1	La aplicación lee las consultas a realizar desde la base de datos local SQLite.
	2	El sistema devuelve las consultas a realizar contra el servidor.
	3	La aplicación realiza las consultas contra el servidor.
	4	El sistema informa de la realización de las consultas con éxito.
Post condición	Se actualizan los resultados del paciente que haya realizado esas sesiones en el servidor.	
Alternativa o Flujo Alternativo	Paso	Alternativa o Flujo Alternativo
	4	Si alguna de las consultas devuelve un error por duplicado, se vuelve al paso 3 actualizando el número de sesión.
	4	Si el servidor no responde con un error el caso de uso finaliza sin llegar a la post condición.
Requisitos Especiales	Ninguno	
Frecuencia esperada	Media	
Importancia	Media	
Urgencia	PD	
Comentarios		

Tabla 8: Especificaciones caso de uso realizar consultas

4.3.9 Obtener actualización de tratamiento

Identificador	Obtener actualización de tratamiento	
Versión	1.0	
Fecha de la última revisión	11/06/2018	
Autores	Marcos Muñoz Morales	
Descripción	La aplicación actualiza la información del tratamiento almacenado para el paciente activo asignado al dispositivo.	
Personal involucrado o Actores	EntrenaTu	
Precondición	Debe haber conexión con el servidor. Debe haber un paciente activo asignado al dispositivo.	
Escenario principal de éxito o	Paso	Escenario principal de éxito o Secuencia o Flujo Normal



Secuencia o Flujo Normal	1	La aplicación consulta contra el servidor con la información del paciente y del tratamiento almacenado.
	2	El sistema devuelve la información del tratamiento actualizada.
	3	La aplicación almacena la información del tratamiento en su base de datos local.
	4	El sistema informa de la realización de la consulta con éxito.
Post condición	Se actualiza la información del tratamiento asignado al paciente del dispositivo.	
Alternativa o Flujo Alternativo	Paso	Alternativa o Flujo Alternativo
	3	Si el sistema devuelve algún error al realizar la consulta se finaliza sin llegar a la post condición.
Requisitos Especiales	Ninguno	
Frecuencia esperada	Media	
Importancia	Media	
Urgencia	PD	
Comentarios		

Tabla 9: Especificaciones caso de uso obtener actualización de tratamiento

4.4 REQUISITOS NO FUNCIONALES

Un requisito funcional o atributo de calidad es la descripción de una propiedad que debe de tener el sistema o una restricción que debe de respetar. Si los funcionales indican que hace el sistema los no funcionales indican cómo de bien lo hace: Disponibilidad, usabilidad, seguridad, rendimiento, etc.

RNF-1. El sistema deberá ser escalable y soportar la concurrencia de usuarios sin importar la magnitud.

RNF-2. La autenticación de terapeutas se realizará mediante usuario y contraseña.

RNF-3. El sistema deberá tener una disponibilidad de 24 horas al día y 7 días a la semana.

RNF-4. La base de datos del servidor debe permitir su modificación sin afectar a la lógica de negocio.

RNF-5. El sistema debe ofrecer un análisis de la sesión realizada.

RNF-6. El sistema deberá incluir lengua española, inglesa, francesa y portuguesa.

RNF-7. El sistema deberá ofrecer un tiempo bajo de respuesta a las consultas realizadas contra el servidor.



RNF-8. El sistema deberá ofrecer un diseño responsivo de modo que se adapte a las pantallas de los diferentes dispositivos.

RNF-9. El sistema deberá evitar la publicación de información que pueda ser considerada inapropiada ética o moralmente.

RNF-10. El sistema deberá proteger la información tanto personal como de la cuenta de cada usuario no permitiendo su publicación.

4.5 REQUISITOS DE INFORMACIÓN

Se tratan de todos aquellos requisitos que describen las restricciones del sistema en relación a los datos almacenados o procesados por el mismo.

RI-1. El sistema almacenara los datos del paciente activo del dispositivo (nombre, apellidos, foto, centro, clave, tratamiento).

RI-2. El sistema almacenara los datos del tratamiento del paciente activo del dispositivo (identificador, ejercicios contenidos, duración, número de sesión, número de ciclo, información tratamiento-paciente)

RI-3. El sistema utilizará el idioma por defecto del usuario ya sea paciente o terapeuta, de no ser posible, utilizará el idioma por defecto de la aplicación (español).



5. MANUAL DE USUARIO

Este capítulo pretende explicar las posibles interacciones que puede tener un usuario con la aplicación. Se atenderán a los dos posibles usuarios a los va dirigida: pacientes y terapeutas.

La aplicación está diseñada para estimular diferentes modalidades cognitivas, las cuales se explicarán de forma independiente al igual que cada pantalla desarrollada, explicando el funcionamiento de tipo de ejercicio que puede ser ejecutado y su desarrollo o funcionamiento.

5.1 ICONO DE LA APLICACIÓN

La aplicación se representa con el icono de acceso a la misma, este es común sin importar la plataforma sobre la que nos encontremos.



Ilustración 18: Icono aplicación EntrenaTu

Se muestra el nuevo icono de EntrenaTu, se ha decidido utilizar un icono diferente que en la versión ya desarrollada de GRADIOR a fin de evitar confusiones. Se utiliza una Tablet a fin de representar el nuevo carácter de movilidad que quiere darse a la aplicación.

5.2 MODALIDADES

El programa se basa en la ejecución de actividades para estimular las diferentes funciones cognitivas en base a niveles de habilidad y ejecución (Antem & Orta, 2017).

El programa incluye ejercicios cognitivos de distintas modalidades.

5.2.1 Atención

La atención es un proceso que elige los estímulos que se captan. A la hora de recibir información es importante los elementos a los que prestamos atención. Para el desarrollo de esta habilidad cognitiva se distinguen los siguientes ejercicios:

- Atención selectiva visual secuencial: El paciente debe indicar en que momentos aparece un nuevo elemento por pantalla, estos elementos aparecen de forma secuencial. Nunca aparecen dos elementos al mismo tiempo. En niveles altos se añade un estímulo distractor el cual es una respuesta errónea al ejercicio.
- Atención selectiva visual simultánea: El paciente debe indicar que imágenes de las que ve son iguales a la de un modelo que se le muestra. Al paciente se le muestran numerosos estímulos de forma simultánea. El paciente es el encargado de indicar que estímulos son los que cumplen con la afirmación.
- Atención vigilancia color: El paciente observa unos elementos situados en pantalla, estos cambian de color cada cierto periodo de tiempo. El paciente únicamente debe pulsar el botón de responder cuando el color coincide con el indicado en el enunciado del ejercicio.
- Atención vigilancia destellos: Se le muestra una imagen de fondo al sujeto, este fondo va cambiando de color cada periodo de tiempo. El paciente únicamente debe hacer clic sobre el fondo cuando este cumple la condición de la afirmación previa del ejercicio.



- Atención sostenida color: Similar a atención vigilancia color, aunque con unos tiempos de respuesta mayores haciendo que el paciente deba mantener durante un tiempo mayor la atención. El paciente observa unos elementos situados en pantalla, estos cambian de color cada cierto periodo de tiempo. El paciente únicamente debe pulsar el botón de responder cuando el color coincide con el indicado en el enunciado del ejercicio.
- Atención sostenida destellos: Similar a atención vigilancia destellos, aunque con unos tiempos de respuesta mayores haciendo que el paciente deba mantener durante un tiempo mayor la atención. Se le muestra una imagen de fondo al sujeto, este fondo va cambiando de color cada periodo de tiempo. El paciente únicamente debe hacer clic sobre el fondo cuando este cumple la condición de la afirmación previa del ejercicio.
- Atención sostenida figuras: Similar a atención selectiva visual secuencial, aunque con unos tiempos de respuesta mayores haciendo que el paciente deba mantener durante un tiempo mayor la atención. El paciente debe indicar en que momentos aparece un nuevo elemento por pantalla, estos elementos aparecen de forma secuencial. Nunca aparecen dos elementos al mismo tiempo. En niveles altos se añade un estímulo distractor el cual es una respuesta errónea al ejercicio. Memoria

5.2.2 Orientación

La orientación es el conjunto de funciones psíquicas que nos permiten darnos cuenta de la situación real en la que nos encontramos. Para el desarrollo de estos ejercicios se pregunta al paciente por el momento actual en el que vive: momento del día, año actual, estación del año, mes, día de la semana y día del mes.

5.2.3 Calculo

El cálculo es una actividad cognitiva que se realiza para relacionar cosas. Esta modalidad pretende entrenar la necesidad de hacer cálculos con todo lo que encontramos a nuestro alrededor. Para este tipo de ejercicios se tienen las siguientes sub-modalidades:

- Calculo cuantitativo contar: Se muestran una serie de estímulos como pueden ser imágenes de animales, objetos o personas. El paciente debe ser capaz de responder a la pregunta que se le realizará previamente o tras la visualización de los estímulos según el nivel de dificultad, esta pregunta siempre estará relacionada con el número de veces que aparece uno o varios de esos elementos.
- Calculo identificación de números: Se le muestran al paciente unos estímulos numéricos de forma simultánea. El paciente debe ser capaz de reconocer cuales de estos estímulos cumplen la condición que se le indica, estas condiciones pueden ser: número más alto, número más bajo, números pares, números impares, o números múltiplos de un determinado valor. Todo ello depende del nivel de dificultad.
- Problemas aritméticos: Esta sub-modalidad de ejercicios trata exactamente de lo que indica su nombre. Se le muestra y reproduce un problema aritmético al sujeto, y este debe ser capaz de resolverlo mentalmente e indicar la respuesta correcta. De no ser capaz de realizarlo, finalmente se le muestran las posibles soluciones y el sujeto debe elegir una en un nuevo intento.



5.2.4 Razonamiento

Con el razonamiento el paciente es capaz de realizar funciones relacionadas con la lógica, estrategia y planificación. Es posible adquirir nuevos conocimientos a través de los ya conocidos a través de llevar a diferentes conclusiones con un mismo trabajo.

Para el desarrollo de esta habilidad se utilizan ejercicios como son:

- Razonamiento ordenar gráfica: En esta sub-modalidad de ejercicios el paciente debe ordenar los estímulos visuales según la afirmación que se le muestra. Por ejemplo, ordenar los vehículos de más rápido a más lento, o elementos de más nuevo a más antiguo. El paciente debe razonar el porqué de la respuesta antes de seleccionarla de forma aleatoria.
- Razonamiento ordenar textos: Similar a la anterior sub-modalidad, en este caso en lugar de utilizar estímulos en forma de imágenes se utilizan estímulos en forma de texto. Se le pide al paciente que ordene de forma razonada los pasos para la realización de un proceso. Son procesos simples como puede ser escribir una carta.

5.2.5 Percepción

La percepción es el proceso que tiene la función de codificar y coordinar las diferentes sensaciones y darles un significado. Los estímulos que vemos llegan a nuestros receptores, los receptores envían información a nuestro cerebro y una vez el cerebro recibe la información debe procesarla. Para el desarrollo de esta habilidad se utilizan ejercicios como son:

- Percepción colores auditiva: En este tipo de ejercicios se ven una serie de estímulos visuales por pantalla. Estos estímulos en niveles bajos se encuentran de forma inmóvil, aunque según aumentan los niveles toman un movimiento que cambia en sentido y velocidad según este asignado. El paciente únicamente debe hacer clic sobre los elementos que cumplen la condición de color indicada al comienzo del ejercicio de forma auditiva. El paciente debe permanecer atento en la previa del ejercicio para ser consciente de que color es importante tocar y durante el transcurso del ejercicio debe ser capaz de distinguir los diferentes colores que aparecen por pantalla para solo seleccionar el color correcto.
- Percepción colores gráfica: Ejercicios similares a percepción colores auditiva, la variación de este ejercicio con el anterior se basa en la forma en que se informa de que estímulos son los correctos. En este caso un rectángulo superior contiene cuadros con el color o colores correctos pintados.
- Percepción colores texto: Ejercicios similares a percepción colores auditiva y percepción colores gráfica, la variación de este ejercicio con el anterior se basa en la forma en que se informa de que estímulos son los correctos. En este caso un rectángulo superior contiene cuadros con el color o colores correctos en formato de texto.
- Percepción visual figuras: Son ejercicios similares a los de la sub-modalidad atención selectiva visual simultánea, en este caso puede ser necesario indicar que imágenes son exactamente iguales al modelo o por el contrario, que imágenes tienen relación con este sin llegar a ser exacto. Esta relación puede ser, por ejemplo, que alimentos o vehículos pertenecen a la misma familia.
- Percepción visual rostros: Son ejercicios similares a los de la sub-modalidad atención selectiva visual simultánea, en este caso puede ser necesario indicar que imágenes son exactamente iguales al modelo o por el contrario, que imágenes



tienen relación con este sin llegar a ser exacto. Esta relación puede ser, por ejemplo, que rostros pertenecen al mismo sujeto del modelo, estos rostros muestran diferentes expresiones, pero al fin de al cabo son de la misma persona.

- Percepción visual tamaños: En este caso se muestran una cantidad menos de estímulos por pantalla que en los anteriores. Únicamente se debe hacer clic en el elemento que cumpla la condición de tamaño (cual es más grande o cual es más pequeño). Esta afirmación puede estar referida al tamaño por pantalla o al tamaño en el mundo real, se debe tener cuidado con ello.

5.2.6 Lenguaje

El lenguaje se entiende por una forma especial de memoria que permite recordar la relación entre una palabra y su significado. Para el entrenamiento de esta habilidad se utilizan estas sub-modalidades de ejercicios:

- Lenguaje comprensión de palabras: Esta sub-modalidad se trata de ejercicios, en los cuales, se muestran unos estímulos de forma gráfica como pueden ser imágenes de animales u objetos. A través de un estímulo auditivo se indica sobre cuál de los dos estímulos debe hacerse clic.
- Lenguaje reconocimiento de palabras: Sub-modalidad similar a la anterior, en este caso en lugar de utilizar estímulos gráficos en forma de imágenes se utilizan estímulos en forma de texto sobre los que se debe hacer clic siguiendo el mismo procedimiento que en el caso anterior.
- Lenguaje identificación de letras escritas: En esta sub-modalidad se muestra un estímulo tanto visual como sonoro, en función del nivel será una letra, una sílaba o una palabra completa. Una vez mostrada y escuchada se pasa a la siguiente pantalla de ejecución, donde el paciente debe indicar la letra que apareció o indicar letra a letra la palabra aparecida.
- Lenguaje identificación de letras oral: Sub-modalidad exactamente igual que la anterior. La única diferencia que se encuentra es la falta del estímulo visual previo de la letra, sílaba o palabra. Toda la información que tiene el sujeto se recibe de forma sonora.

5.2.7 Función ejecutiva

Estas capacidades son las necesarias para hacer funcionar correctamente la memoria, juntando la información almacenada o planificar adecuadamente unos recuerdos. Son las encargadas de poner en marcha, organizar e integrar el resto de las habilidades cognitivas de nuestro cerebro. Para el desarrollo de esta modalidad se utilizan los siguientes tipos de ejercicios:

- Cambio reglas: Se trata de ejercicios en los cuales se muestra una condición y se debe indicar de forma rápida si el elemento visual que vemos la cumple o no. Esta condición puede estar relacionada solamente con el estímulo visual que se ve en ese momento o con ese y algún estímulo visto anteriormente.
- Claves: Se muestran dos tablas, una tabla superior y una tabla inferior, en la tabla superior se muestra cada estímulo visual con su clave, esta clave puede ser numérica o gráfica. En la tabla inferior únicamente se muestran las claves, ya sean ordenadas o desordenadas. En la parte central de la pantalla se irán mostrando de forma secuencial diferentes estímulos de los contenidos en la parte superior de la pantalla, y se debe indicar a que clave corresponde cada uno de ellos haciendo clic en la tabla inferior.



- **Inhibición auditiva:** Se mostrará una condición tanto de forma visual como sonora. Al paciente se le irán reproduciendo diferentes audios ya sean letras (en niveles bajos) o palabras completas (en niveles altos), y el paciente únicamente debe hacer clic en el botón pulsar cuando se cumpla la condición de la secuencia de audio que se le presenta.
- **Inhibición visual:** Semejante al ejercicio de inhibición auditiva con una serie de diferencias. En este caso esos estímulos sonoros son estímulos visuales que tienen un movimiento animado por pantalla. El paciente en lugar de tener que hacer clic sobre un botón debe hacer clic sobre el estímulo visual que cumple la condición antes de que complete su recorrido.
- **Interferencia:** En estos ejercicios se combinan dos tipos de estímulos, un elemento en forma de texto con una palabra y un elemento interferencia visual, este elemento interferencia puede ser una imagen, un color o una serie de puntos. Únicamente se debe hacer clic en el elemento que cumpla la condición.
- **Números y letras:** Se trata de un ejercicio que se combina con memoria y razonamiento. Se muestra unas secuencias de números y letras desordenada. Posteriormente esta secuencia se debe ordenar de forma alfanumérica con los números al principio o final según indique la frase de ejecución.
- **Ordenación historias:** Se muestra una secuencia desordenada de la realización de una actividad (ir a una cafetería, poner la mesa, escribir una carta, etc.). El paciente debe hacer clic en las imágenes en función del orden en el que se debe realizar esa secuencia.
- **Puzles:** Al paciente se le muestra una imagen en miniatura de un modelo, y esa misma imagen en un tamaño superior a la que le faltan determinadas piezas. El paciente debe hacer clic en las piezas y en los huecos que crea que encajen para completar el puzle.

5.2.8 Memoria

La memoria es un proceso o conjunto de procesos que permiten codificar, almacenar y recuperar la información una vez ha sido recogida por nuestras neuronas. Existen varios tipos de memoria que se puede entrenar, inmediata, corto plazo, largo plazo, implícita... Para realizar el desarrollo de todos estos tipos de memoria se distinguen las siguientes sub-modalidades de ejercicios:

- **Memoria auditiva corto plazo:** En estos ejercicios se reproduce de forma sonora una serie de estímulos auditivos, el paciente debe tratar de memorizar los que pueda. Tras esta reproducción se intenta distraer al sujeto con una breve pantalla de distracción. Seguidamente de forma secuencial se pide al paciente que indique si las palabras que van apareciendo estaban o no en la secuencia que se reprodujo previamente.
- **Memoria auditiva inmediata:** Similar a memoria auditiva corto plazo, pero sin esa pantalla de distracción, cambiando ese tipo de memoria de corto plazo a memoria inmediata.
- **Memoria auditiva largo plazo:** Similar a memoria auditiva corto plazo, pero en este caso en lugar de intercalar una pantalla de distracción pueden intercalarse ejercicios completos entre la reproducción de los estímulos y la pantalla de ejecución. Estos ejercicios pueden configurarse para que duren 1 minuto o todos los minutos que se quiera, siempre dentro de la sesión en la que se encuentre el paciente.



- Memoria gráfica corto plazo: Similar a memoria auditiva corto plazo, la diferencia en este caso es que los estímulos se muestran de forma visual y la pantalla de ejecución no muestra los elementos de forma secuencia, si no, de forma simultánea, debiendo hacer clic solo sobre los elementos vistos en la secuencia previa.
- Memoria gráfica corto plazo compuesta: Similar a memoria auditiva corto plazo, la diferencia en este caso es que los estímulos se muestran de forma visual, pero se mantiene la pantalla de ejecución con los estímulos apareciendo de forma secuencial.
- Memoria gráfica inmediata: Similar a memoria gráfica corto plazo, eliminando la pantalla intermedia de distracción para el uso de la memoria inmediata.
- Memoria gráfica largo plazo: Similar a memoria gráfica corto plazo, eliminando la pantalla intermedia de distracción, sustituyéndola como en el caso de memoria auditiva largo plazo por ejercicios completos para el uso de la memoria a largo plazo.
- Memoria verbal corto plazo: Similar a memoria gráfica corto plazo, pero sustituyendo los estímulos visuales por la combinación de estímulos sonoros y visuales de palabras.
- Memoria verbal corto plazo compuesta: Similar a memoria gráfica corto plazo compuesta, pero sustituyendo los estímulos visuales por la combinación de estímulos sonoros y visuales de palabras.
- Memoria verbal inmediata: Similar a memoria gráfica inmediata, pero sustituyendo los estímulos visuales por la combinación de estímulos sonoros y visuales de palabras.
- Memoria verbal largo plazo: Similar a memoria gráfica largo plazo, pero sustituyendo los estímulos visuales por la combinación de estímulos sonoros y visuales de palabras.
- Memoria implícita: Similar a memoria verbal corto plazo, pero cambiando la pantalla de ejecución, se muestran varios estímulos de forma simultánea, pero únicamente uno de esos elementos es una respuesta correcta.
- Memoria localización: Se muestran unos elementos visuales situados por pantalla como marca el ejercicio, el paciente debe recordar la posición de todos ellos, tras un tiempo se muestra una pantalla de distracción y a continuación se muestra una pantalla similar a la primera, pero con los elementos invisibles. El paciente únicamente debe hacer clic en los huecos donde se encontraban los elementos que se piden.
- Memoria semántica: Son ejercicios con preguntas acerca de conocimientos adquiridos a lo largo de años, preguntas comunes de eventos importantes sucedidos en los últimos años o conocimientos básicos. Se reproduce una frase de ejecución y se muestran de forma simultánea las posibles soluciones siendo únicamente una la solución correcta.
- Memoria span letras directa: Se muestra una secuencia de letras la cual el paciente debe recordar. Posteriormente en la pantalla de ejecución el paciente debe indicar letra a letra la secuencia vista previamente.
- Memoria span letras inversa: Se muestra una secuencia de letras la cual el paciente debe recordar. Posteriormente en la pantalla de ejecución el paciente debe indicar letra a letra la secuencia vista previamente, pero en orden inverso.
- Memoria span números directa: Similar a memoria span letras directa, en este caso en lugar de utilizar letras como elementos se utilizarán números.



- Memoria span números inversa: Similar a memoria span letras inversa, en este caso en lugar de utilizar letras como elementos se utilizarán números.
- Memoria span objetos directa: Similar a memoria span letras directa, en este caso en lugar de utilizar letras como elementos se utilizarán imágenes de objetos.
- Memoria span objetos inversa: Similar a memoria span letras inversa, en este caso en lugar de utilizar letras como elementos se utilizarán objetos.

5.3 MÉTODO CREACIÓN PANTALLA DE EJERCICIOS UNIVERSAL

Todos los ejercicios se encuentran almacenados en formato binario en la base de datos gracias a la tabla ejercicios. En esta tabla se encuentra un campo llamado oEjercicio, este campo almacena la información del ejercicio y de sus pantallas. Si se observara este campo en formato de texto claro, se apreciaría claramente un fichero con formato XML. Pero no necesitamos este fichero en texto claro, sino, convertir ese array binario en una clase mediante una deserialización XML.

```
<?xml version="1.0" ?>
<ejercicios>
<memoriaLibreExercise xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <nombre>LENGRECON_21.2</nombre>
  <descripcion>Se solicita el reconocimiento de dos palabras de uso menos frecuente. Nivel de dificultad: 1</descripcion>
  <localeCode>es-ES</localeCode>
  <duracion>1</duracion>
  <nivelDificultad>2</nivelDificultad>
  <tiempoEspera>15</tiempoEspera>
  <pantallas>
    <pantalla xsi:type="IntroductionScreen">
      <nombre>Introducci?n_1</nombre>
      <colorFondo>White</colorFondo>
      <elementos>
        <elemento xsi:type="PauseButton">
          <elemento xsi:type="TextElement" idElement="3912621595" id="3912621595">
            <colorFondo>Transparent</colorFondo>
            <posicion>
              <X>0.1006</X>
              <Y>0.0503</Y>
            </posicion>
            <dimensiones>
              <ancho>0.8239</ancho>
              <alto>0.8595</alto>
            </dimensiones>
            <colorBorde>Black</colorBorde>
            <estiloBorde>Solid</estiloBorde>
            <anchoBorde>0</anchoBorde>
            <reproducirSonido>true</reproducirSonido>
            <estiloTexto>MiddleCenter</estiloTexto>
            <fuente>Georgia; 30pt; style=Bold</fuente>
            <colorTexto>Black</colorTexto>
            <isReplayTextAtStimulusChange>false</isReplayTextAtStimulusChange>
            <isShowOnlyWhenRepeat>false</isShowOnlyWhenRepeat>
            <canShowHideText>false</canShowHideText>
            <valorFuenteNormalizado valorFuenteNormalizado="0.0625" tipoReferenciaFuente="Alto" />
            <fraseEjecucion>false</fraseEjecucion>
          </elemento>
        </elementos>
        <recogePuntuacion>false</recogePuntuacion>
        <imagenFondo>1720673335</imagenFondo>
        <tiempoEntreSonidos>0.5</tiempoEntreSonidos>
        <seccion>1</seccion>
        <tiempoMinimoVisualiacion>5</tiempoMinimoVisualiacion>
      </pantalla>
    <pantalla xsi:type="FreeSimultaneousExecutionScreen">
    <pantalla xsi:type="FreeSimultaneousExecutionScreen">
  </pantallas>
  <refuerzosAcierto>
  <refuerzosFallo>
  <refuerzosOmision>
  <estimulosCriticos />
  <tipoEjercicio>3</tipoEjercicio>
</memoriaLibreExercise>
</ejercicios>
```

Ilustración 19: Ejemplo fichero XML ejercicio

```
MemoryStream mem1 = new MemoryStream(b);  
Clases.ejercicios ejercicio = (Clases.ejercicios)Clases.utilidades.MyStaticValues.XML.Deserialize(mem1);
```

Ilustración 20: Comandos deserialización XML ejercicio

Tras la deserialización se obtendrá una clase con la información de todas y cada una de las pantallas que componen cada ejercicio. Una vez obtenida esta clase, basta con leer los diferentes elementos que la componen obteniendo identificadores de imágenes o audios, colores, tamaños o posiciones para crear las diferentes pantallas de cada ejercicio con una apariencia similar sin importar el dispositivo o sistema en el que se ejecute.

5.4 PANTALLAS

5.4.1 Pantalla de descarga de contenido

Esta pantalla únicamente aparece en caso de comprobar la propia aplicación que no contiene los mismos elementos que la base de datos del servidor. Pidiendo al usuario que la permita descargar y sustituir los elementos y ejercicios de la base de datos local por los contenidos en ese momento en el servidor.

La pantalla no bloquea la aplicación, pero sí que obliga a la descarga de este contenido si se desea continuar con la ejecución de las siguientes pantallas, en caso contrario, únicamente da la opción de salir de la aplicación.

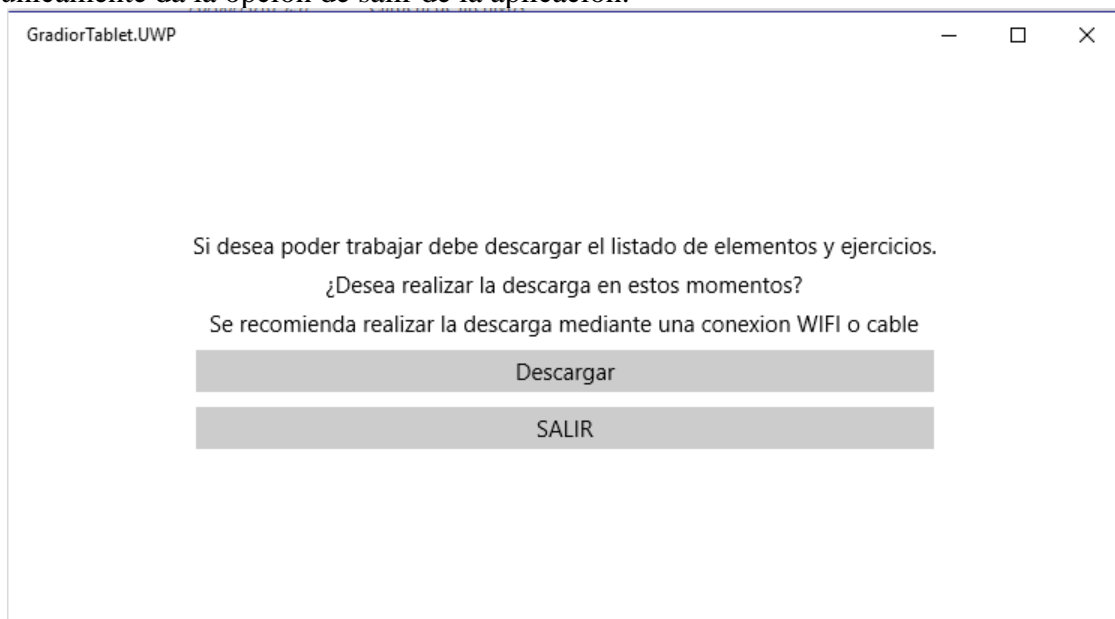


Ilustración 21: Pantalla de descarga de contenido

Para el acceso de esta pantalla no importa si se es paciente o terapeuta, es una pantalla común para ambos.

5.4.2 Pantalla inicio paciente

Esta pantalla es la pantalla principal al abrir la aplicación cualquier usuario, en ella se puede identificar el paciente para iniciar su tratamiento o acceder al terapeuta a su área para su autenticación. La primera vez que se ejecute la aplicación no habrá paciente asignado por lo que el usuario que la inicie observará el siguiente mensaje:

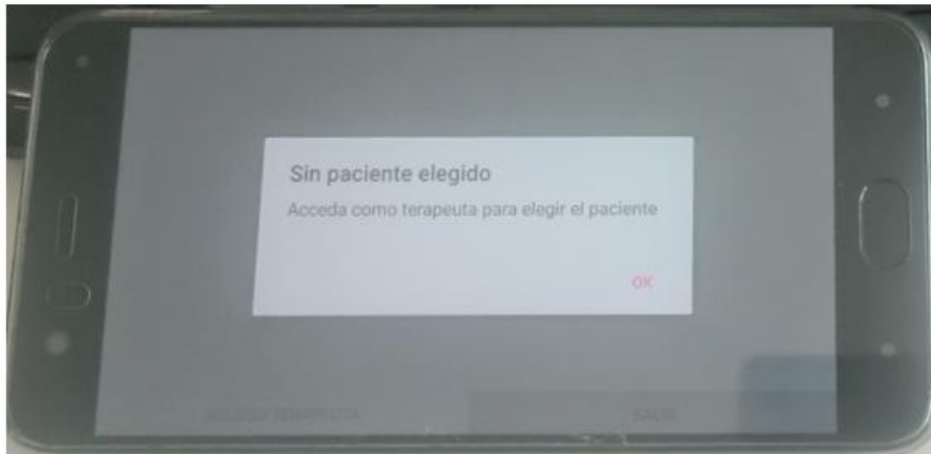


Ilustración 22: Pantalla inicial sin paciente asignado

Será obligatorio que un terapeuta acceda a su área reservada en la aplicación y asigne un paciente con tratamiento activo al dispositivo.



Ilustración 23: Pantalla inicial con paciente asignado

5.4.3 Pantalla autenticación terapeuta

El terapeuta puede configurar el paciente hacia el que va destinado la aplicación, para ello debe acceder a su área reservada de configuración.

Para el acceso serán necesarias sus credenciales (usuario y contraseña). En caso de ser incorrectas se notificará de ello por pantalla.

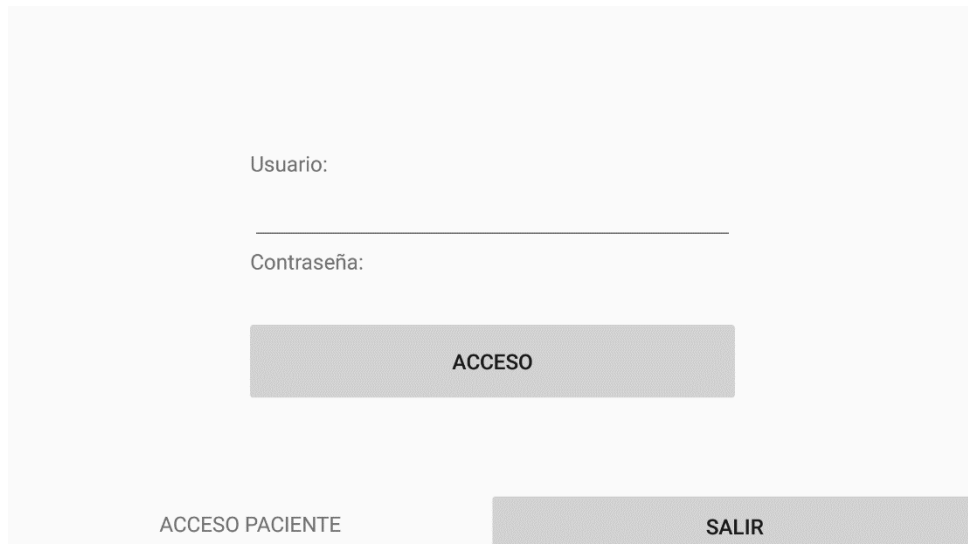


Ilustración 24: Pantalla autenticación terapeuta

5.4.4 Pantalla de asignación de paciente activo

Una vez autenticado el terapeuta, tiene dos opciones:

- Asignar paciente activo: Para indicar el paciente activo basta con seleccionar de la lista superior el paciente que se desea asignar a la aplicación y hacer clic en el botón “UTILIZAR COMO PACIENTE ACTIVO Y VOLVER”. Una vez realizado esto, se volverá a la pantalla de inicio de la aplicación.
- Borrar paciente activo: Basta con pulsar el botón y el paciente registrado como activo de la aplicación se borrará, volviendo al estado por defecto sin paciente.

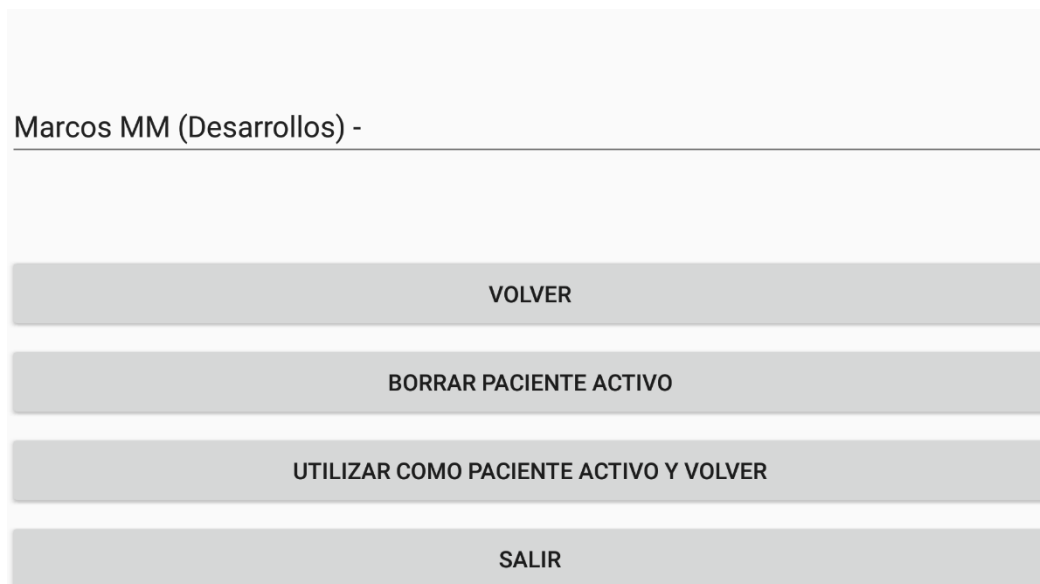


Ilustración 25: Pantalla asignación de paciente activo

5.4.5 Pantalla de estadísticas

Al finalizar la realización de cualquier sesión de tratamiento por parte del paciente se muestra una pantalla con un *feedback* hacia este. Esta retroalimentación está diseñada para mejorar la reacción del paciente con el resultado de sus ejercicios, haciéndole ver su

mejoría sin ser necesaria la presencia del terapeuta de forma presencial para informarle de sus resultados.

Se muestran los resultados de cada modalidad de forma independiente haciendo ver cuáles son los posibles puntos fuertes o débiles del paciente, e intentando incentivarle para mejorar en aquellas en las que puede estar más flojo.



Ilustración 26: Pantalla de estadísticas

5.4.6 Pantalla de introducción

Se trata de una pantalla común para la casi totalidad de los ejercicios, en ella se capta la atención del paciente de modo que se concentre en la realización del ejercicio.

Se intenta informar de forma breve que debe realizar el paciente para realizar el ejercicio con éxito, de este modo una vez que comience la ejecución del ejercicio este no pierda un tiempo importante.

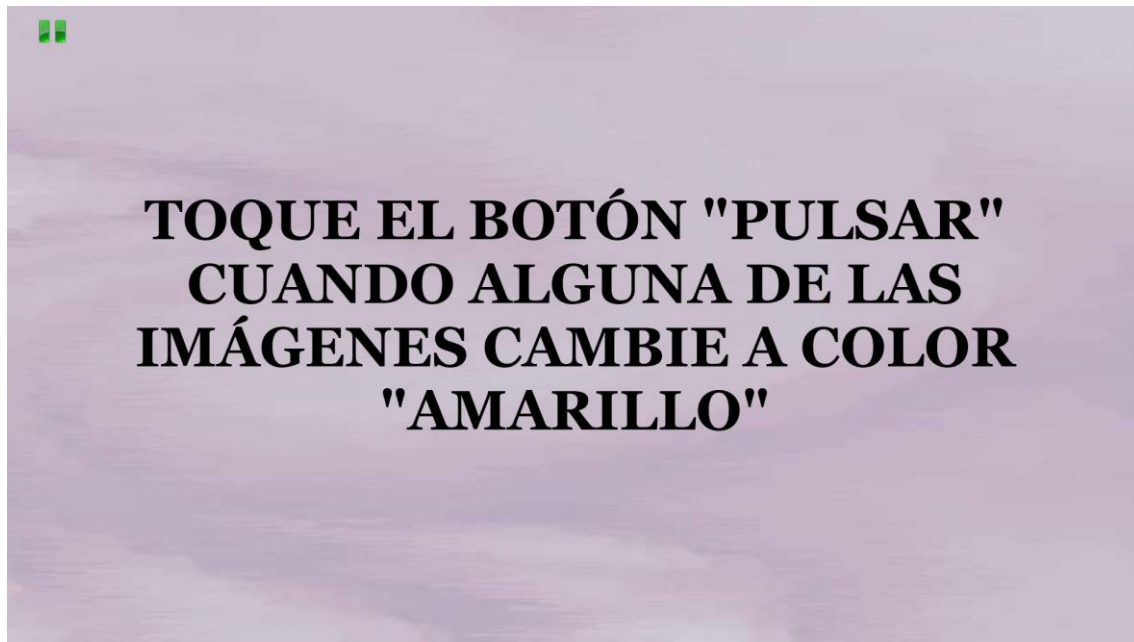


Ilustración 27: Introduction Screen

Esta pantalla también es utilizada en ejercicios en los cuales se dispone de más de un intento para su realización. En el caso de que no se realice de forma correcta se informa al paciente de que tiene un nuevo intento, o de que lo ha realizado con éxito y que pasa a la siguiente sesión, en caso de tener más de una secuencia de ejecución el ejercicio.

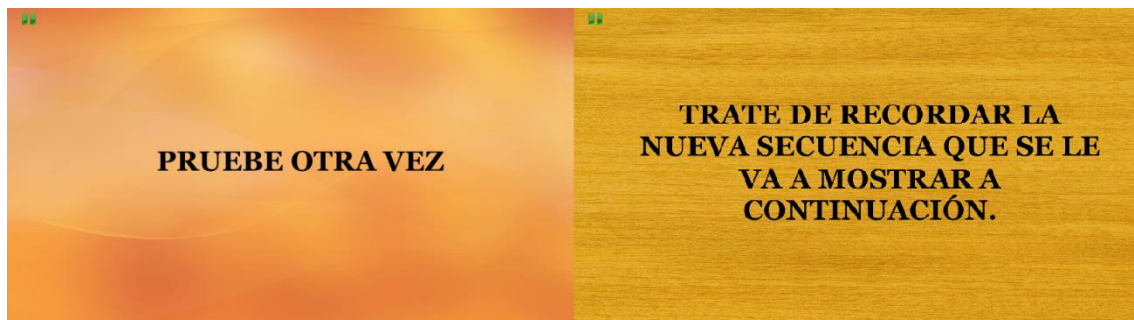


Ilustración 28: Ejemplo pantalla de introducción en caso de fallo o acierto de un ejercicio con varias secuencias

5.4.7 Distraction screen

Vista utilizada en diferentes ejercicios de memoria. Se intercala entre las pantallas del ejercicio una pantalla con un fondo y audio que tratan de hacer perder la concentración al paciente. En esta vista el paciente solo debe esperar a que salte la siguiente pantalla de ejecución.



Ilustración 29: Distraction screen

5.4.8 Atención Selectiva Visual Execution Screen

Vista utilizada en ejercicios de atención sostenida figuras o atención selectiva visual secuencial. Esta pantalla hace aparecer una serie de estímulos de forma secuencial. El paciente debe hacer clic en el estímulo en el momento en que aparece, de no hacerlo se contabilizará como omisión y de hacerlo en el fondo y no sobre la figura contabilizará como fallo por comisión.



Ilustración 30: Atención Selectiva Visual Execution Screen

5.4.9 Atención vigilancia Color Execution Screen

Vista utilizada para los ejercicios de atención vigilancia color, atención vigilancia destellos, atención sostenida color y atención sostenida destellos. Existen dos tipos de ejecuciones para estas pantallas:

- Color: Se muestran unos elementos que van cambiando, el paciente debe hacer clic en el botón una vez el estímulo cambia al color especificado por la pantalla de introducción.



Ilustración 31: Atención vigilancia color o atención sostenida color

- Destellos: El fondo de pantalla es el propio botón, cuando el fondo de pantalla cambia al color indicado se deberá hacer clic sobre él

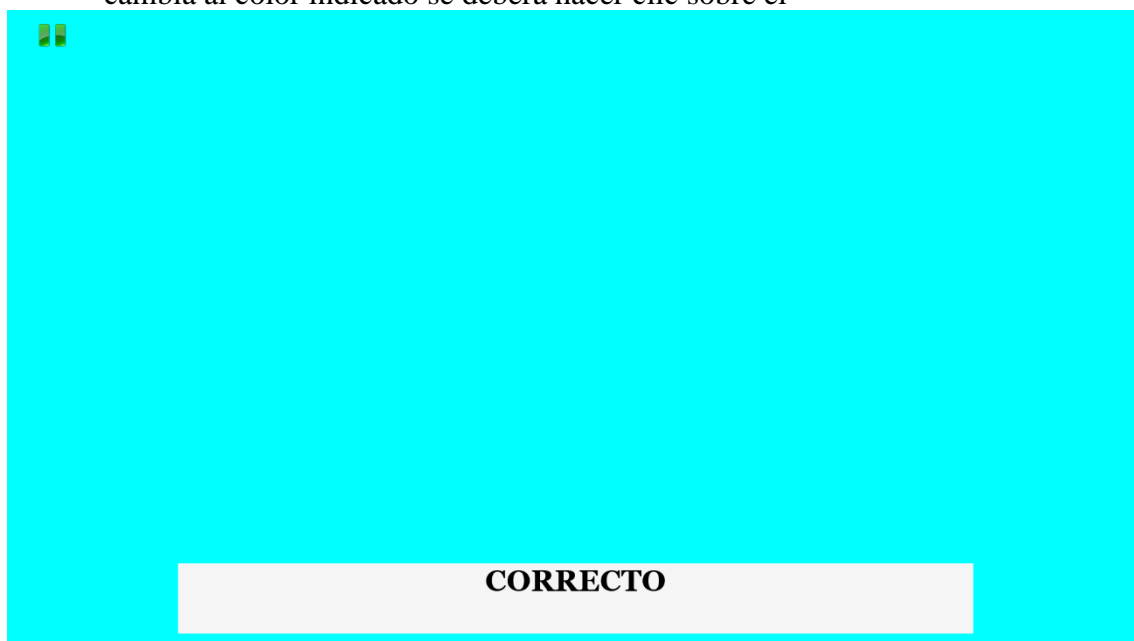


Ilustración 32: Atención vigilancia destellos o atención sostenida destellos

5.4.10 Ejecutiva cambio reglas si no execution screen

Vista utilizada para los ejercicios de función ejecutiva cambio reglas. Se muestra una instrucción que va cambiando, el paciente únicamente debe hacer clic en el botón si o botón no según el estímulo que aparezca en ese turno, el cual va cambiando de forma secuencial.



Ilustración 33: Ejecutiva cambio reglas si no execution screen

5.4.11 Ejecutiva tarea claves sequential execution screen

Vista utilizada para los ejercicios de función ejecutiva claves. Se muestra una tabla superior e inferior, en la tabla superior se muestra cada estímulo de forma individual con un número u objeto asignado como respuesta. Ese número u objeto se forma del mismo modo en la parte inferior de la pantalla ya sea ordenado o desordenado según el nivel. El paciente debe hacer clic en la respuesta que coincida con el estímulo que se muestra en la parte central de la pantalla.

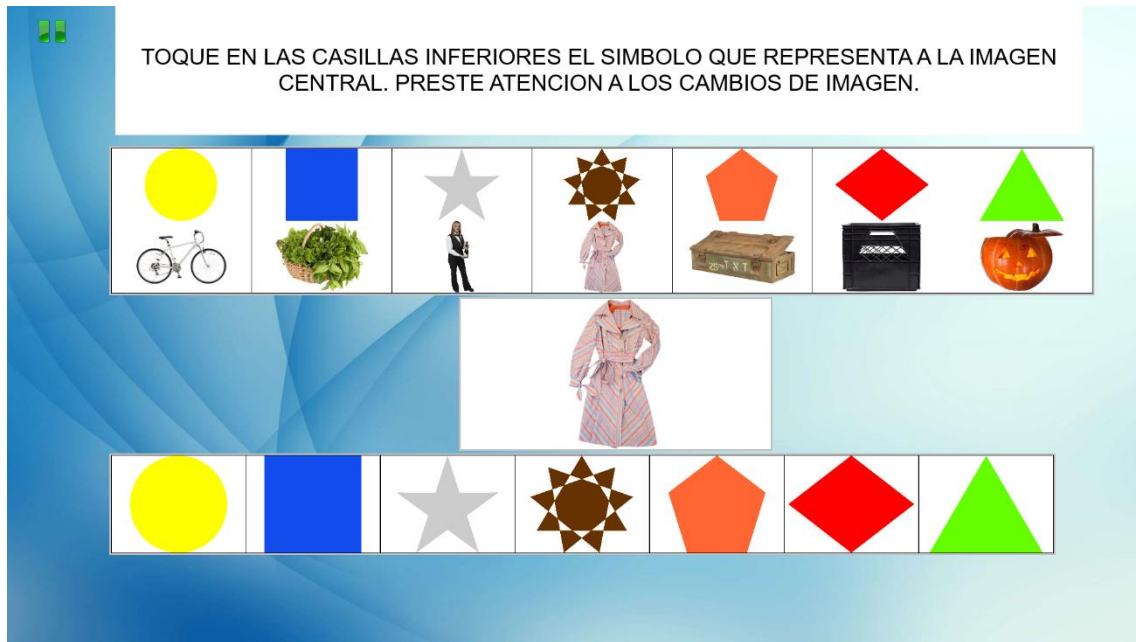


Ilustración 34: Ejecutiva tarea claves sequential execution screen

5.4.12 Puzzle execution screen

Vista utilizada para los ejercicios de función ejecutiva puzles. Se muestra un modelo con la imagen original y esa misma imagen ampliada habiendo borrado partes del puzle de modo que el paciente debe hacer clic en el hueco y en las piezas inferiores para reconstruirlo. El orden en el que se forme no afecta al resultado final.



Ilustración 35: Puzzle execution screen

5.4.13 Ejecutiva ordenación execution screen

Vista utilizada para los ejercicios de función ejecutiva ordenación historias. Se muestran unas imágenes, estas imágenes bien ordenadas forman una historia, haciendo clic sobre estas imágenes se irán ordenando. De no acertar la siguiente imagen del turno en el número de intentos asignados se mostrara la corrección de forma automática.

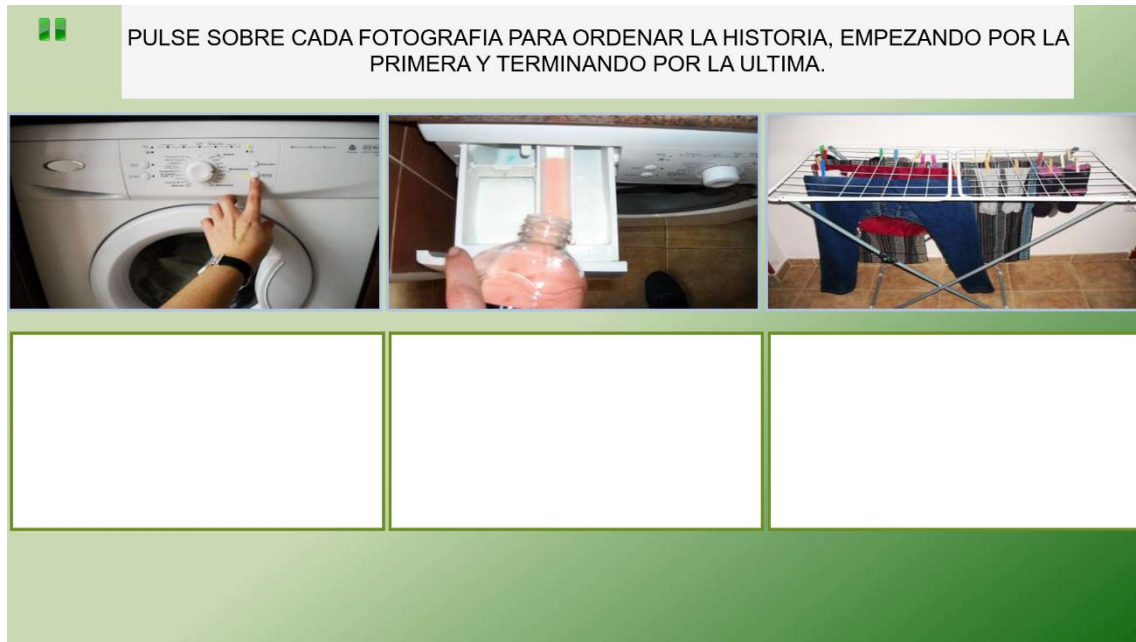


Ilustración 36: ejecutiva ordenación execution screen

5.4.14 Ejecutiva tarea números y letras presentation screen

Vista utilizada para los ejercicios de función ejecutiva números y letras. Se muestra una secuencia de números y letras de forma secuencial desordenada de modo que el paciente la recuerde.

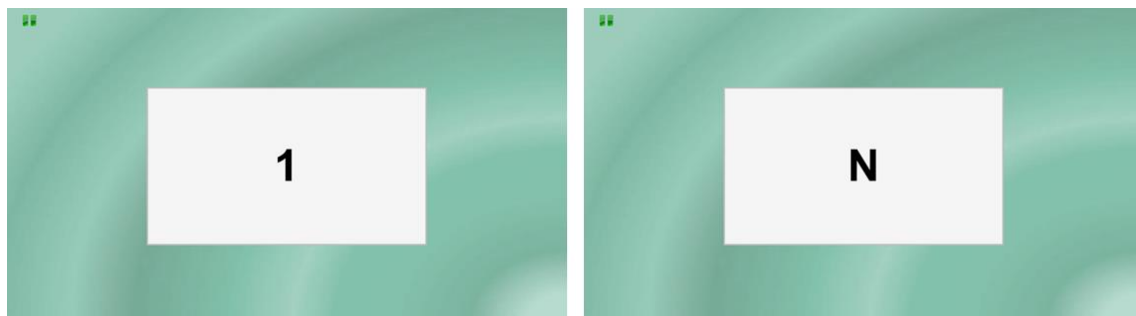


Ilustración 37: Ejecutiva tarea números y letras presentation screen

5.4.15 Span sequential presentation Screen

Vista utilizada en ejercicios de lenguaje identificación de letras escritas u orales en niveles 2 y 3, memoria span de cualquier tipo, memoria corto plazo de cualquier tipo, memoria inmediata de cualquier tipo, memoria largo plazo de cualquier tipo, memoria corto plazo compuesta de cualquier tipo y memoria implícita. La pantalla es exactamente igual añadiendo la posibilidad de presentar objetos o únicamente audios sin verse el texto.



Ilustración 38: Span sequential presentation screen

5.4.16 Ejecutiva interferencias simultaneous execution screen

Vista utilizada para los ejercicios de función ejecutiva interferencia en niveles bajos. Se muestran unos estímulos y una frase de ejecución. Se debe hacer clic sobre los estímulos oportunos. De no hacerlo de forma correcta se mostrara la solución.



Ilustración 39: Ejecutiva interferencias simultaneous execution screen

5.4.17 Ejecutiva interferencia si no execution screen

Vista utilizada para los ejercicios de función ejecutiva interferencia en niveles altos. Semejante a la pantalla anterior pero en lugar de mostrarse las opciones de forma simultánea se muestran de forma secuencial. El paciente debe hacer clic en sí o no según sea o no correcta la afirmación.



Ilustración 40: Ejecutiva interferencia si no execution screen

5.4.18 Ejecutiva inhibición auditiva execution screen

Vista utilizada para los ejercicios de función ejecutiva inhibición auditiva. Se reproducen de forma secuencia audios de letras o palabras y el paciente debe hacer clic en el botón cuando se cumpla la condición de la afirmación.



Ilustración 41: Ejecutiva inhibición auditiva execution screen

5.4.19 Ejecutiva inhibición execution screen

Vista utilizada para los ejercicios de función ejecutiva inhibición visual. Se muestra una secuencia de imágenes en movimiento. El paciente debe hacer clic sobre las imágenes que cumplan la condición antes de que desaparezcan de pantalla.



Ilustración 42: Ejecutiva inhibición execution screen

5.4.20 Solución problemas presentational screen

Vista utilizada en ejercicios de cálculo problemas aritméticos y cálculo cuantitativo contar niveles bajos. Se presenta de forma visual y auditiva el problema o la instrucción para la correcta realización del cálculo del ejercicio. En algunos ejercicios, no en todos, es posible avanzar esta pantalla haciendo clic en el botón continuar.

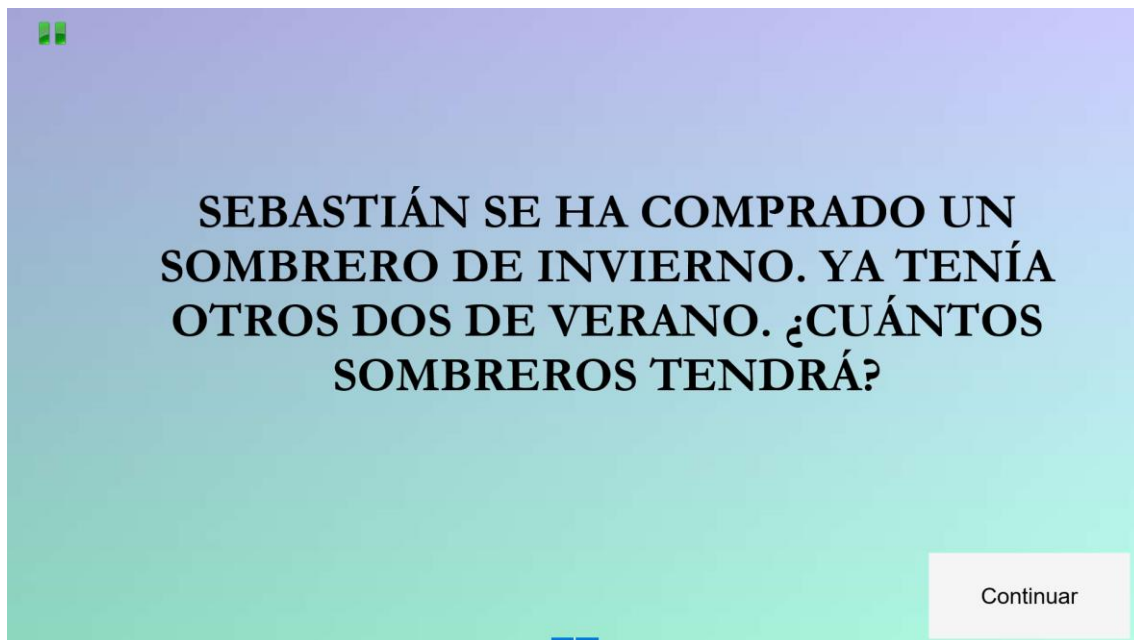


Ilustración 43: Solución problemas presentational screen

5.4.21 Solución problemas execution screen

Vista utilizada en ejercicios de cálculo problemas aritméticos y cálculo cuantitativo contar niveles bajos. Se muestra un panel numérico donde el paciente debe introducir la respuesta correcta. Una vez introducida tras un tiempo se comprueba. En algunos

ejercicios de este tipo es posible hacer clic en un botón de comprobación para agilizar el ejercicio.

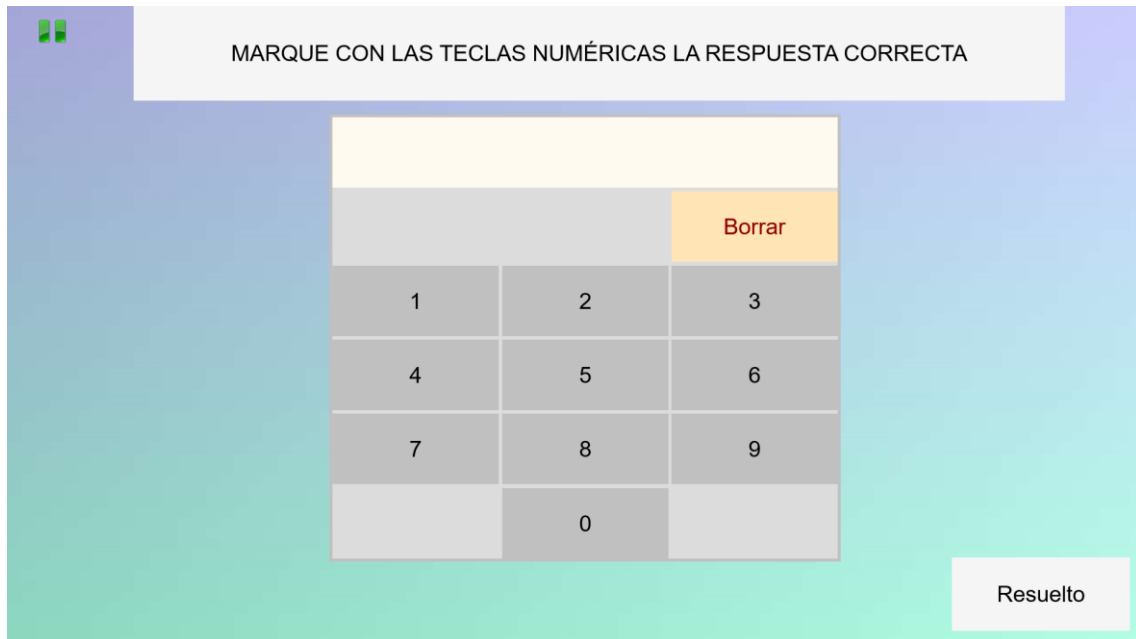


Ilustración 44: Solución problemas execution screen

5.4.22 Solución problemas alternative screen

Vista utilizada en ejercicios de cálculo problemas aritméticos y cálculo cuantitativo contar niveles bajos. En esta pantalla se pueden dar dos casos, o bien se resuelve el ejercicio, o bien se dan una serie de opciones que el paciente debe escoger. En caso de que falle esta opción se muestra la solución final del problema.

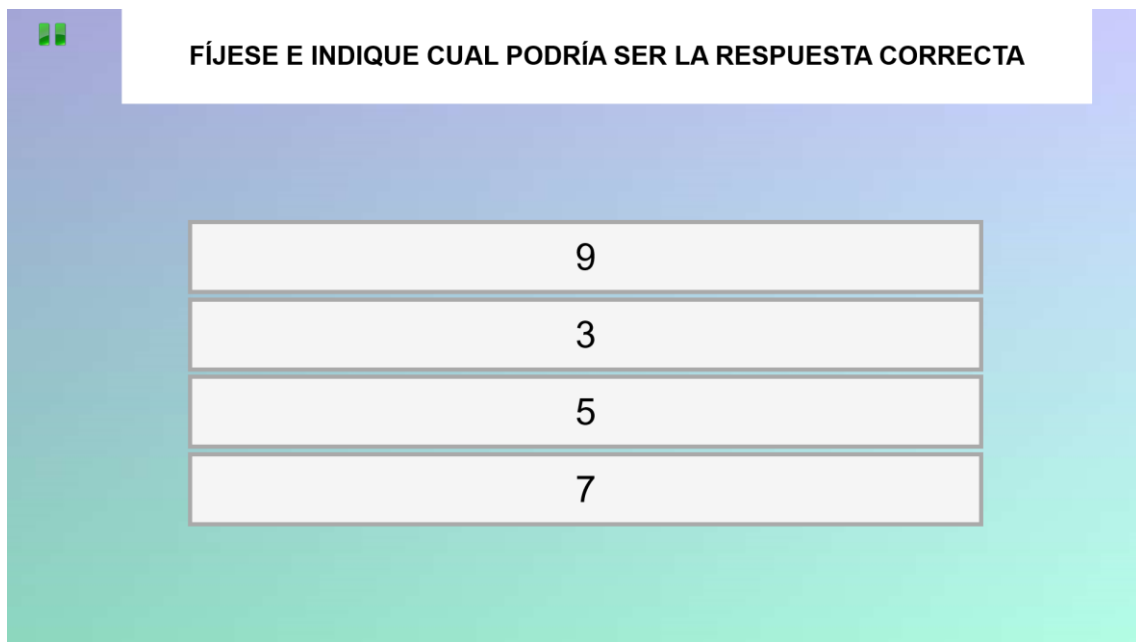


Ilustración 45: Solución problemas execution screen

5.4.23 **Ejecutiva tarea contar presentation screen**

Vista utilizada en ejercicios de cálculo cuantitativo contar niveles altos. La pantalla es similar a la utilizada en niveles bajos, añadiéndose una serie de animaciones en los elementos que dificultan el ejercicio al paciente.



Ilustración 46: Ejecutiva tarea contar presentation screen

5.4.24 **Ejecutiva tarea contar execution screen**

Vista utilizada en ejercicios de cálculo cuantitativo contar niveles altos. Su funcionamiento y apariencia es exactamente la misma que la pantalla solución problemas execution screen vista en puntos anteriores.

5.4.25 **Ejecutiva tarea contar alternative screen**

Vista utilizada en ejercicios de cálculo cuantitativo contar niveles altos. Solo se accede a esta pantalla en caso de no acertar en la pantalla anterior. En esta pantalla se resuelve el ejercicio mostrándose la solución.



Ilustración 47: Ejecutiva tarea contar alternative screen

5.4.26 Memoria simultaneous execution screen

Vista utilizada en multitud de ejercicios: Calculo identificación de números, lenguaje comprensión de palabras, lenguaje reconocimiento de palabras, memoria corto plazo de cualquier tipo, memoria inmediata de cualquier tipo, memoria largo plazo de cualquier tipo, memoria corto plazo compuesta de cualquier tipo y memoria semántica. En esta pantalla se muestran las posibles opciones bien colocadas en pantalla, y debe hacerse clic en los estímulos que cumplan la condición de la afirmación que se reproduce.



Ilustración 48: Memoria simultaneous execution screen

5.4.27 Memoria sequential execution screen

Vista utilizada en ejercicios de memoria corto plazo de cualquier tipo, memoria inmediata de cualquier tipo, memoria largo plazo de cualquier tipo, memoria corto plazo compuesta.

Se trata de una pantalla que muestra los estímulos de forma secuencial, dando opción de pulsar el botón sí o no según la condición que ponga el ejercicio, o de si esa imagen, palabra o audio se vio en la pantalla de presentación previa a ella.

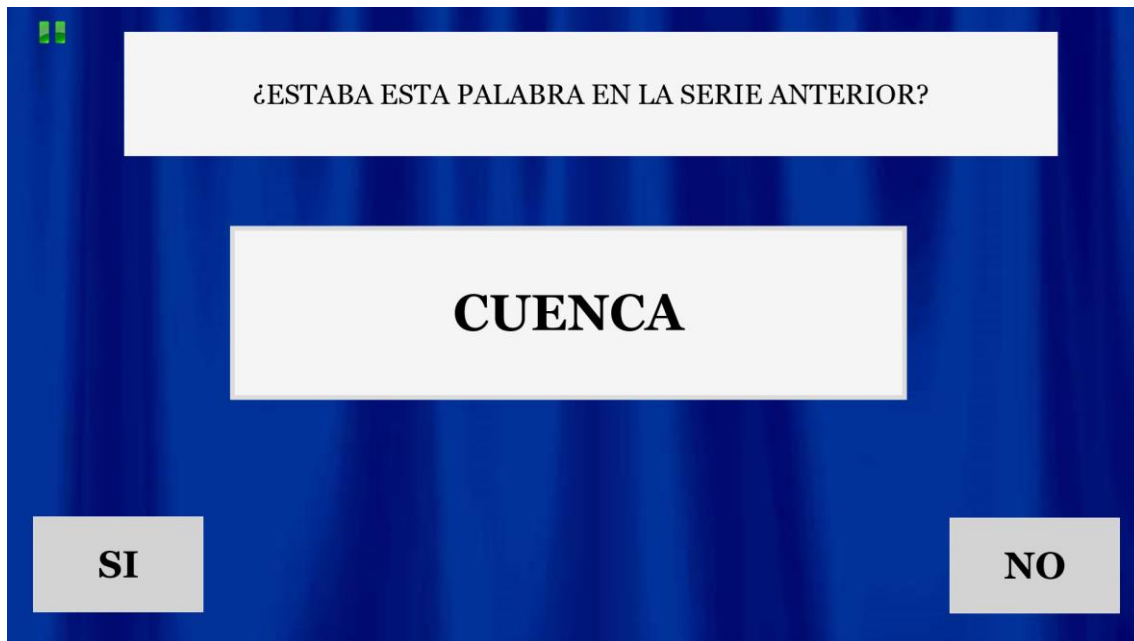


Ilustración 49: Memoria sequential execution screen

5.4.28 Implicit execution screen

Vista utilizada en ejercicios de lenguaje identificación letras escritas u orales nivel 2 o 3 y memoria implícita. En esta pantalla se da una serie de opciones como respuesta correcta, en caso de responder de forma errónea o por omisión se visualiza la opción correcta. Puede funcionar tanto con palabras completas como letra a letra de la palabra mostrada previamente.

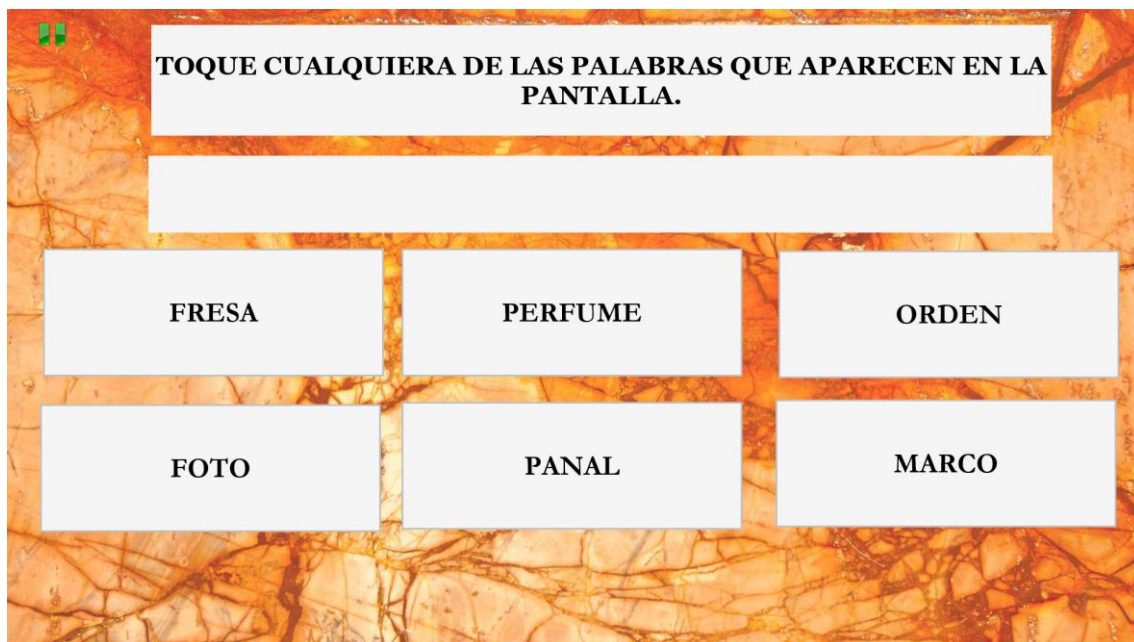


Ilustración 50: Implicit execution screen

5.4.29 **Secuencias execution screen**

Vista utilizada en los ejercicios de las sub-modalidades de función ejecutiva números y letras, lenguaje identificación letras escritas u oral de nivel 1 y memoria span de cualquier tipo. En esta pantalla se pueden mostrar letras, números, u objetos para que se ordenen en el orden directo o inverso que se vieran previamente. O incluso ordenados de forma alfabéticamente con opción de números antes o después según lo pida el ejercicio función ejecutiva números y letras.

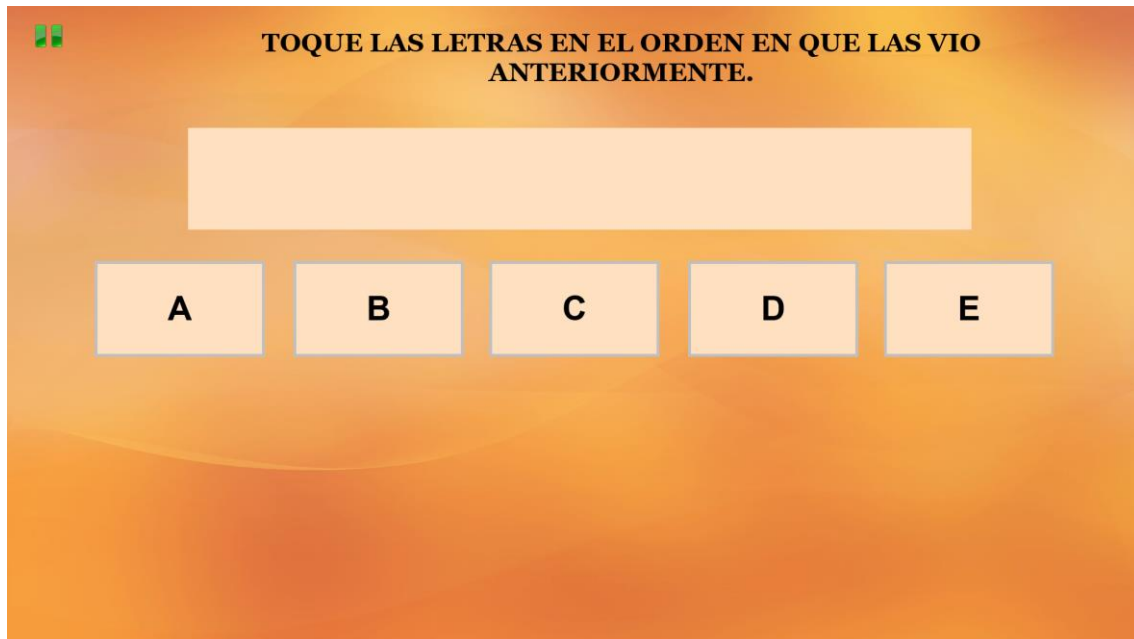


Ilustración 51: Secuencias execution screen

5.4.30 **Percepción color execution screen**

Vista utilizada en los ejercicios de percepción colores auditiva, gráfica o texto. En esta pantalla se muestran los estímulos especificados en la construcción del ejercicio debiendo hacer clic el paciente en los estímulo oportunos. Pueden mostrarse tanto de forma estática inmóviles como con un movimiento animado que recorre la pantalla a mayor o menos velocidad según el nivel en el que se encuentre el paciente.

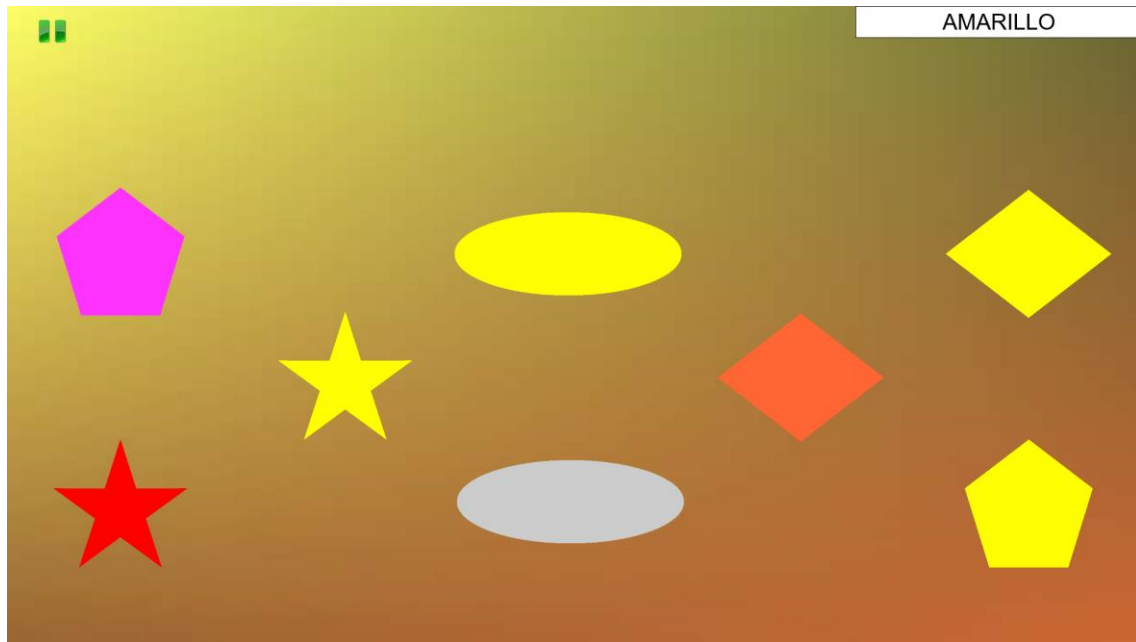


Ilustración 52: Percepción color execution screen

5.4.31 Percepción visual execution screen

Vista utilizada en los ejercicios de atención selectiva visual simultánea o percepción visual figuras, rostros o tamaños. Se muestran unos estímulos de forma simultánea en pantalla, el paciente debe hacer clic sobre los que cumplan la condición, algunos de estos ejercicios mostrarán un modelo de referencia en la parte superior de la pantalla.



Ilustración 53: Percepción visual execution screen

5.4.32 Razonamiento ordenar execution screen

Vista utilizada para los ejercicios de las modalidades de razonamiento ordenación gráfica o razonamiento ordenación textos. En estos ejercicios se muestran unos estímulos sobre los que se debe hacer clic en el orden que indique la condición.

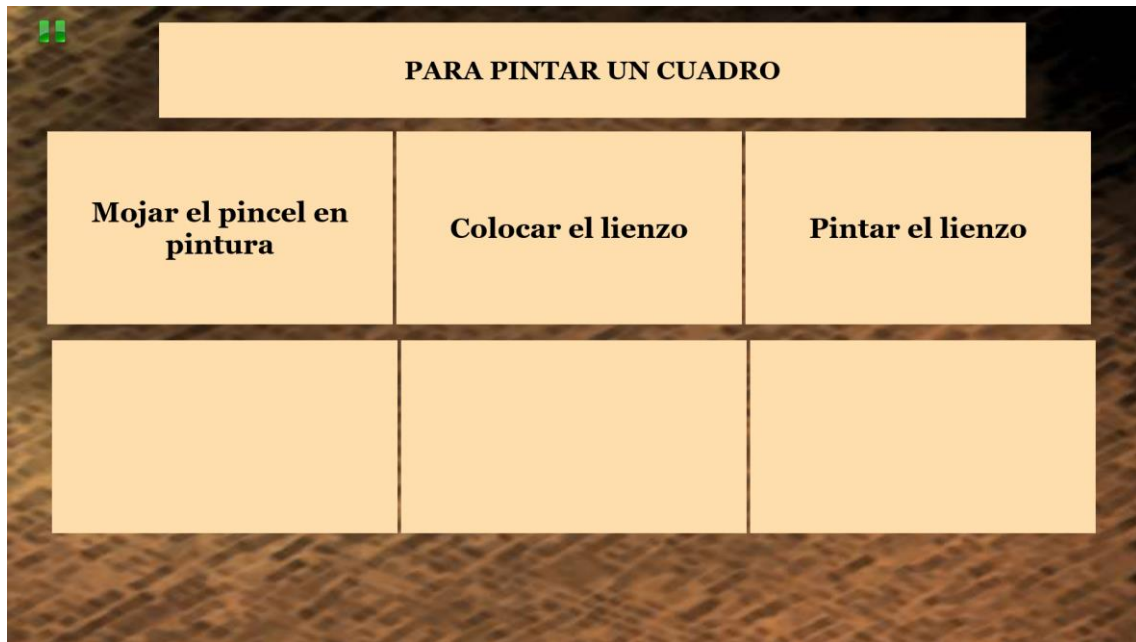


Ilustración 54: Razonamiento ordenar execution screen

5.4.33 Simultaneous presentation screen

Vista utilizada para la modalidad de memoria localización, en ella se representan imágenes. El paciente debe recordar donde se encuentra cada imagen.

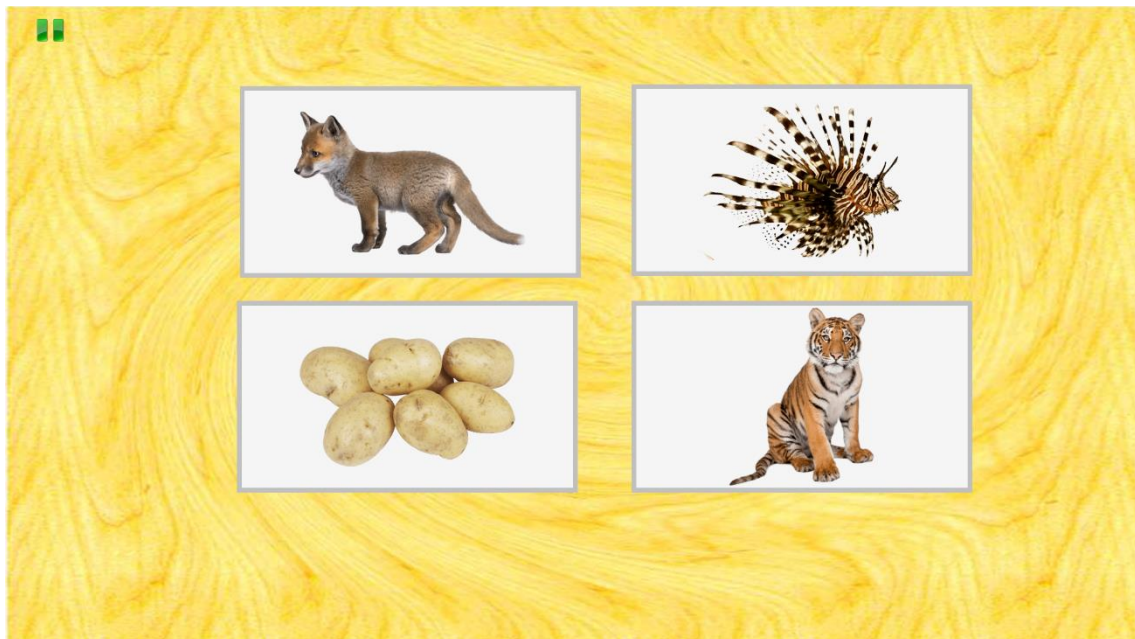


Ilustración 55: Simultaneous presentation screen

5.4.34 Simultaneous execution screen

Vista utilizada para la modalidad de memoria localización. Es la pantalla de ejecución que sigue a la vista anterior. En ella el paciente debe hacer clic sobre el hueco que coincida con la afirmación que se le indica. Entre ambas pantallas se intercala una pantalla de distracción.



Ilustración 56: Simultaneous execution screen

5.4.35 Momento día execution screen

Vista utilizada en los ejercicios de orientación. En ella se pide al usuario que indique en que momento del día se encuentra en su localización. De no responder de forma correcta se pueden tener más intentos si así está indicado o se puede pasar a la siguiente pantalla contabilizando el error y mostrando la solución.



Ilustración 57: Momento día execution screen

5.4.36 Year execution screen

Vista utilizada en los ejercicios de orientación. En ella se pide al usuario que indique el año actual. De no responder de forma correcta se pueden tener más intentos si así está indicado o se puede pasar a la siguiente pantalla contabilizando el error y mostrando la

solución. Esta pantalla tiene dos tipos de ejecuciones, puede ser un panel numérico y el paciente debe introducirlo, o pueden mostrarse opciones y el paciente elegir una.



Ilustración 58: Year execution screen

5.4.37 Estaciones 4 execution screen

Vista utilizada en los ejercicios de orientación. En ella se pide al usuario que indique la estación actual según su localización. De no responder de forma correcta se pueden tener más intentos si así está indicado o se puede pasar a la siguiente pantalla contabilizando el error y mostrando la solución.



Ilustración 59: Estaciones 4 execution screen

5.4.38 Día semana execution screen

Vista utilizada en los ejercicios de orientación. En ella se pide al usuario que indique el día de la semana actual. De no responder de forma correcta se pueden tener más intentos si así está indicado o se puede pasar a la siguiente pantalla contabilizando el error y mostrando la solución.

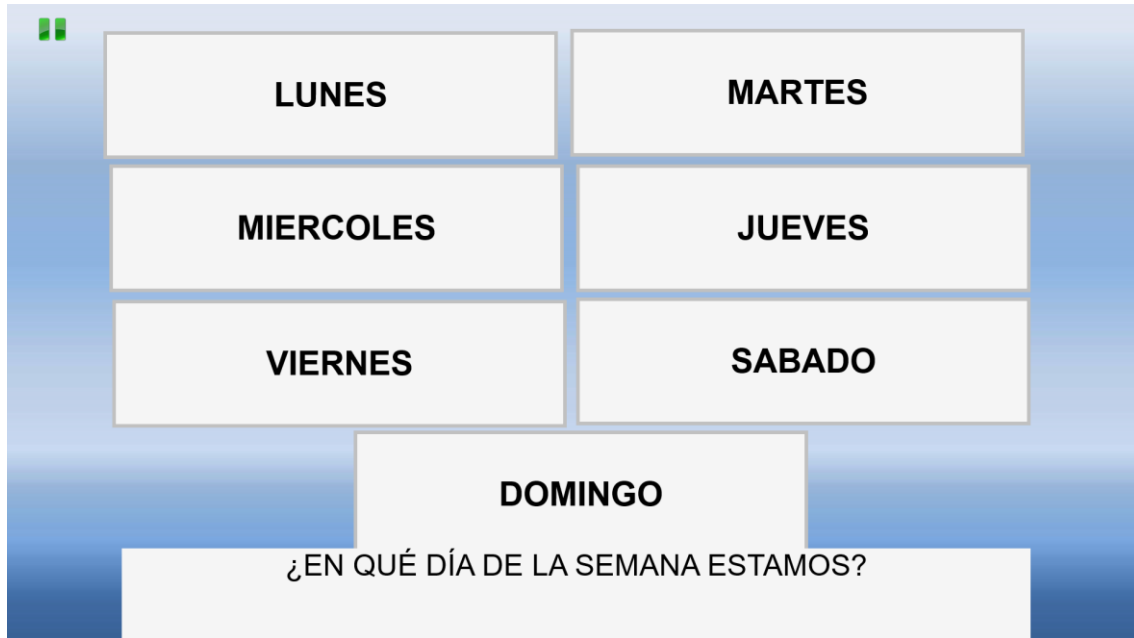


Ilustración 60: Día semana execution screen

5.4.39 Día mes execution screen

Vista utilizada en los ejercicios de orientación. En ella se pide al usuario que indique el día del mes actual. De no responder de forma correcta se pueden tener más intentos si así está indicado o se puede pasar a la siguiente pantalla contabilizando el error y mostrando la solución.



1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

¿QUÉ DÍA ES HOY?

Ilustración 61: Día mes execution screen

Una vez respondida esta pantalla se muestra una solución global de los ejercicios de orientación.

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

HOY ES... 22 ESTAMOS EN EL MES DE JUNIO DEL AÑO 2018
CAE EN VIERNES

Ilustración 62: Día mes execution screen solución final





6. CONCLUSIONES

A lo largo de este proyecto se han investigado y estudiado las diferentes tecnologías para el desarrollo de aplicaciones móviles (aplicaciones nativas, aplicaciones web, aplicaciones híbridas y aplicaciones generadas). Eligiendo finalmente la opción que por coste y tiempo de desarrollo mejor se adaptaba a los intereses de la Fundación, optando por una aplicación generada multiplataforma a través de la novedosa API de Xamarin.

La aplicación desarrollada se constituye como un sistema para el apoyo a terapeutas en la rehabilitación cognitiva de pacientes. El enfoque multiplataforma logra no limitar en exceso el dispositivo destino hacia el que se enfoca. Logrando llegar tanto a dispositivos móviles como en algunos casos a equipos de escritorio con sistema operativo Windows 10. Este abanico de dispositivos permite que un paciente pueda realizar un tratamiento en un entorno más familiar o más relajado sin tener que estar presente en una clínica mediante un dispositivo del que podía ya disponer o adquirirlo de forma sencilla y que no tiene que utilizar exclusivamente para sus tratamientos en todo momento.

La aplicación, a través de las clases que la forman, sirve de motor lanzador de ejercicios traduciendo el código del que se componían los ejercicios en estímulos visuales, sonoros. La propia aplicación se adapta en función del turno establecido de los ejercicios y distingue entre las diferentes pantallas de ejecución existentes, de este modo, los ejercicios se realizan de forma clara y ordenada según lo establecido por el terapeuta en el diseñador de tratamientos del que dispone. De cara al paciente se intenta que el funcionamiento de los ejercicios sea sencillo, utilizando las funciones táctiles que permiten los dispositivos móviles a fin de poder realizar los ejercicios de forma más intuitiva, haciendo que sepan usar todos los recursos de la misma.

Una de las mayores ventajas que presenta es la no dependencia exclusiva de la conexión a internet. Basta una primera conexión para descargar todos los elementos y ejercicios necesarios en los tratamientos, y posteriormente, la asignación de un paciente activo en el dispositivo por parte del terapeuta. Una vez cumplidos los requisitos mencionados, el paciente, puede ejecutar las sesiones de su tratamiento asignado tantas veces como el terapeuta se lo indique, y en caso de no disponer de conexión con el servidor en el momento de guardar los resultados, estos se almacenan localmente hasta disponer de conexión para el envío y análisis en el servidor de la clínica.

Destacar que se han alcanzado de forma satisfactoria los objetivos que se iban presentando conforme se avanzaba en el desarrollo de la aplicación. El reto que supone comenzar el aprendizaje de una tecnología novedosa con no demasiados desarrollos que tomar como referencia, como es Xamarin. Siendo necesario realizar el aprendizaje de forma autónoma y la dificultad de resolución de algunos de los problemas surgidos en el manejo de diferentes tipos de objetos que a base de diferentes búsquedas en la comunidad de Microsoft se resolvieron satisfactoriamente.





7. LÍNEAS FUTURAS

El sistema desarrollado únicamente está enfocado hacia la realización de tratamientos de forma remota por parte de los pacientes de la clínica. Existiendo multitud de funcionalidades que es posible añadir o modificar para mejorar esta aplicación en el futuro.

Una opción futura clara a tener en cuenta es la de mejorar el apartado visual de las pantallas no asignadas a ejercicios, haciendo una interfaz gráfica más atractiva y vistosa.

Otra idea de línea futura de desarrollo podría ser el añadir clases que mejoraran de alguna forma las funcionalidades de la aplicación., algunas de las ideas podrían ser:

- Al tratarse de una aplicación móvil, es posible cerrarla sin inconveniente ninguno mientras el transcurso de cualquier ejercicio. Sería interesante la posibilidad de almacenar el estado de ese último ejercicio. De este modo, el paciente en lugar de iniciar una nueva sesión por cerrarse la aplicación debido a algún error de la aplicación o del propio paciente, podría continuar la sesión de tratamiento en la que se encontrase como si nada hubiera sucedido.
- El desarrollo de nuevas sub-modalidades de ejercicios, así como nuevos niveles de dificultad o una modificación en un mismo ejercicio de forma que no sea solo dinámica la elección de los ejercicios si no también la realización de un mismo ejercicio en diferentes momentos. De este modo se reduciría en mayor medida el efecto del aprendizaje de los ejercicios que invalida la rehabilitación del paciente.

En cuanto a los usuarios hacia los que va dirigida la aplicación también podrían realizarse gran cantidad de mejoras futuras como podrían ser:

- Dar más funcionalidades a los terapeutas, la aplicación está diseñada de forma que la única función del terapeuta es la de asignar o no un paciente activo a la aplicación. Una buena idea sería la de permitirle nuevas funciones de las que ya dispone el software GRADIOR para equipos de escritorio, como es el análisis de resultados, la creación de informes o la asignación de tratamientos.
- Un nuevo sistema de autenticación hacia pacientes, previamente los pacientes utilizaban los equipos de la clínica, teniendo a un usuario especializado presente en todo momento que comprueba que el paciente realiza su tratamiento. Al realizar los tratamientos desde un equipo remoto se puede dar el caso de un paciente que realice o intente acceder a un tratamiento de un paciente que no es quien dice ser. Podría modificarse el sistema de autenticación de clave con imágenes de modo que el paciente no deba elegir entre 8 imágenes diferentes en un solo intento, si no, desarrollar un nuevo sistema que compruebe que es quien dice ser realmente.
- Un sistema de notificaciones en caso de existir actualizaciones de elementos o ejercicios, que permitan al dueño del dispositivo realizar la descarga de estos a fin de tener en todo momento su base de datos actualizada.
- Dar la posibilidad a los terapeutas de gestionar de forma online los tratamientos de los pacientes, sin ser necesario instalar el software GRADIOR y de este modo poder realizarlo sin la instalación de software adicional, únicamente con un navegador sería posible realizarlo.
- Añadir nuevas funcionalidades al modo de lanzamiento de tratamientos, como por ejemplo, añadiendo una distinción como es la de modo centro y modo paciente. De este modo se distinguiría la localización del paciente, de estar en su domicilio



se indicaría previamente hacia que paciente va dirigida la aplicación, pero si se encuentra en la clínica, se podría dar la posibilidad de mostrar todos los pacientes disponibles y que sea el terapeuta quien guie al paciente, de este modo no sería necesario editar la configuración cada vez que un nuevo paciente tuviera que ejecutar sus tratamientos.



8. BIBLIOGRAFÍA

Bankinter. (2015). La población de España dentro de 50 años. Recuperado el 26 de junio de 2018, de <https://blog.bankinter.com/economia/-/noticia/2015/01/02/poblacion-espanola-en-2064.aspx>

El Mundo. (2012). El deterioro cognitivo comienza antes de los 45 años. Recuperado el 26 de junio de 2018, de <http://www.elmundo.es/elmundosalud/2012/01/05/neurociencia/1325792191.html>

INTRAS. (2015). Fundación INTRAS. Recuperado el 26 de junio de 2018, de <http://www.intras.es/>

GRADIOR. (2017). Software de Rehabilitación Neuropsicológica del Deterioro Cognitivo (Grador). Recuperado el 26 de junio de 2018, de <http://www.intras.es/index.php/productos/software-gradior>

INTERABLE. (2017). Proyecto Interable: combinación de terapia cognitiva y física. Recuperado el 26 de junio de 2018, de <http://ides.es/blog/proyecto-interable-combinaci%C3%B3n-de-terapia-cognitiva-y-f%C3%ADsica>

Vázquez-Cano, E. & Sevillano García, M. L. (2015). *Dispositivos digitales móviles en educación: El aprendizaje ubicuo*. Madrid: NARCEA, S.A. DE EDICIONES

Statcounter. (2018). Desktop vs Mobile vs Tablet Market Share Worldwide. Recuperado el 26 de junio de 2018, de <http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/#monthly-201401-201805>

Statista. (2016). Tasa de penetración de los sistemas operativos para Smartphone en España desde 2014 hasta 2016. Recuperado el 26 de junio de 2018, de <https://es.statista.com/estadisticas/473759/tasa-penetracion-sistema-operativo-smartphone-espana/>

Cuello, J., Vittone, J. (2013). *Diseñando apps para móviles*. España: appdesignbook

Hébuterne, S. (2016). *Android, Guía de desarrollo de aplicaciones Java para Smartphones y Tabletas*. Barcelona, Ediciones ENI

Arias, A., Flores Gonzalo, E. (2016) *Curso de programación con iOS. Apps Iphone 2º Edición*. España, Createspace Independent Pub

Microsoft (2018). ¿Qué es una aplicación para Plataforma universal de Windows (UWP)? Recuperado el 26 de junio de 2018, de <https://docs.microsoft.com/es-es/windows/uwp/get-started/universal-application-platform-guide>

Gómez, J.L.B. (2016). *Desarrollo de aplicaciones web en el entorno servidor*. Madrod. Ediciones Paraninfo, S.A.



Cabello, A.L.C. (2014). *Implantación de aplicaciones web en entornos internet, intranet y extranet*. Malaga, IC editorial.

Ortuño, N.M. (2016). *Trabajo Fin de Master: Diseño e implementación de una aplicación móvil multiplataforma gestionable por web para la notificación de eventos locales*. Cartagena, Universidad politécnica de Cartagena.

Xamarin Microsoft. (2017). Introducción al desarrollo móvil. Recuperado el 26 de junio de 2018, de <https://docs.microsoft.com/es-es/xamarin/cross-platform/get-started/introduction-to-mobile-development>

Xamarin Forms. (2016). Introducción a Xamarin.Forms. Recuperado el 26 de junio de 2018, de <https://docs.microsoft.com/es-es/xamarin/xamarin-forms/get-started/index>

Antem, J. M. & Orta, G.L. (2017). *Bases neurológicas de las enfermedades cognitivas del desarrollo y su tratamiento educativo*. Madrid, Universidad nacional de educación a distancia

Visual Studio. (2018). Visual Studio. Recuperado el 25 de julio de 2018, de <https://visualstudio.microsoft.com>



ANEXO 1: VISUAL STUDIO

Visual Studio es el entorno de desarrollo integrado oficial para el desarrollo de aplicaciones por parte de Microsoft. Se encuentra disponible tanto para Windows como para Mac, permitiendo el desarrollo de aplicaciones en Android, iOS, Mac, Windows y web (Visual Studio, 2018).

Visual Studio dispone de algunas características enfocadas a la productividad como son:

- Refactorización: cambio inteligente de nombre de variables o movimiento de código a otras ubicaciones.
- IntelliSense: Característica la cual muestra información sobre el código automáticamente.
- Cuadro de búsqueda de inicio rápido: Búsqueda rápida de cualquier información o componente desde Visual Studio.
- CodeLens: Búsqueda de referencias y cambios en el código.

REQUISITOS MÍNIMOS

Es posible la instalación de Visual Studio en los siguientes sistemas operativos de la serie de Microsoft:

- Windows 10 versión 1507 y posteriores: Home, Professional, Education y Enterprise (LTSC y S no se admiten)
- Windows Server 2016: Standard y Datacenter
- Windows 8.1 (con actualización 2919355): Core, Professional y Enterprise
- Windows Server 2012 R2 (con actualización 2919355): Essentials, Standard y Datacenter
- Windows 7 SP1 (con las actualizaciones más recientes de Windows): Home, Premium, Professional, Enterprise y Ultimate
- MacOS Sierra 10.12.
- MacOS X El Capitan 10.11.

Los requisitos mínimos a nivel de hardware son los siguientes:

- Procesador doble núcleo o superior de 1,8GHz o superior.
- 2 GB de RAM, aunque lo recomendado son mínimo 4GB de memoria RAM.
- Hasta 130GB de espacio disponible en el disco duro en función del tipo de instalación que se realice. Para un mejor rendimiento es importante instalar Windows y Visual Studio en una unidad de estado sólido.
- Tarjeta de video con una resolución mínima de 1280x720.

En caso de tratarse de sistema operativo MacOS el requisito mínimo de memoria RAM será de 4GB, siendo recomendados 8GB para su correcto funcionamiento y 1GB en el disco duro.

Un requisito adicional en caso de trabajar desde MacOS sería la necesidad de instalación de .NET Core SDK 1.1, pudiéndose instalar de forma independiente a Visual Studio.

INSTALACIÓN

En primer lugar será necesaria la descarga del instalador de Visual Studio desde la URL: <https://visualstudio.microsoft.com/es/downloads/>.



Ilustración 63: Versiones Visual Studio

En función del desarrollo hacia el que se enfoque su instalación será necesaria una versión diferente. Para el desarrollo en Xamarin basta con la versión gratuita Visual Studio Community 2017, la elección de los paquetes a añadir vendrá a continuación, una vez ejecutado el fichero de instalación. El mínimo de paquetes necesarios para el desarrollo de una aplicación en Xamarin serán los siguientes:

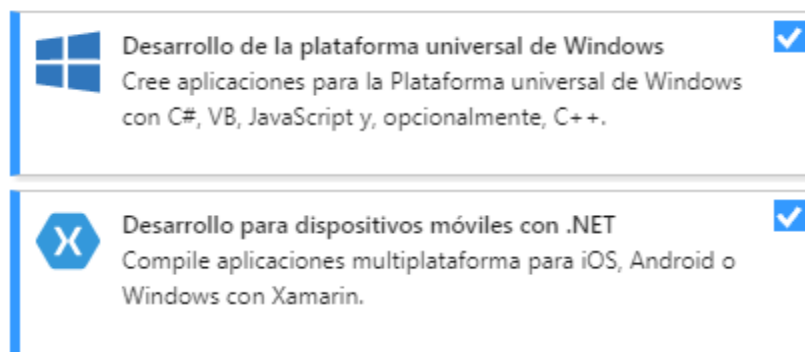


Ilustración 64: Paquetes necesarios desarrollo Xamarin

El paquete de plataforma universal de Windows será necesario para la instalación del SDK de UWP. Algo parecido pasa con el paquete de desarrollo para dispositivos móviles, instalándose el SDK de las plataformas de Android e iOS. De este modo estarán almacenadas todas las librerías para el desarrollo común multiplataforma y en caso de ser necesarias, las librerías independientes de cada una. Estas librerías independientes ayudan en caso de necesitar compilar de forma condicional secciones de código y de este modo acceder a capacidades de los dispositivos que de otro modo no sería posible. La instalación de los emuladores ya depende de cada usuario, pero si se dispone de equipos físicos es posible conectarlos físicamente, o a través de la red, y realizar la depuración directamente en estos.

CREACIÓN DE PRIMER PROYECTO XAMARIN FORMS

Tras iniciar por primera vez el programa se deberán seguir los siguientes pasos para la creación de un proyecto y con ello de las diferentes aplicaciones en cada plataforma.

En primer lugar, se debe crear el proyecto como tal, para ello una vez iniciada la herramienta se hará clic en Archivo > Nuevo > Proyecto...

Una vez realizado el paso anterior, se abrirá una ventana con todas las posibilidades de proyectos disponibles en Visual Studio, tenemos de todo tipo, desde aplicaciones Web a aplicaciones de Escritorio. En este caso se hará clic en Cross-Platform > Aplicación móvil. Se asigna el nombre de proyecto que se crea conveniente, así como el directorio en el que se quiere almacenar los diferentes ficheros.

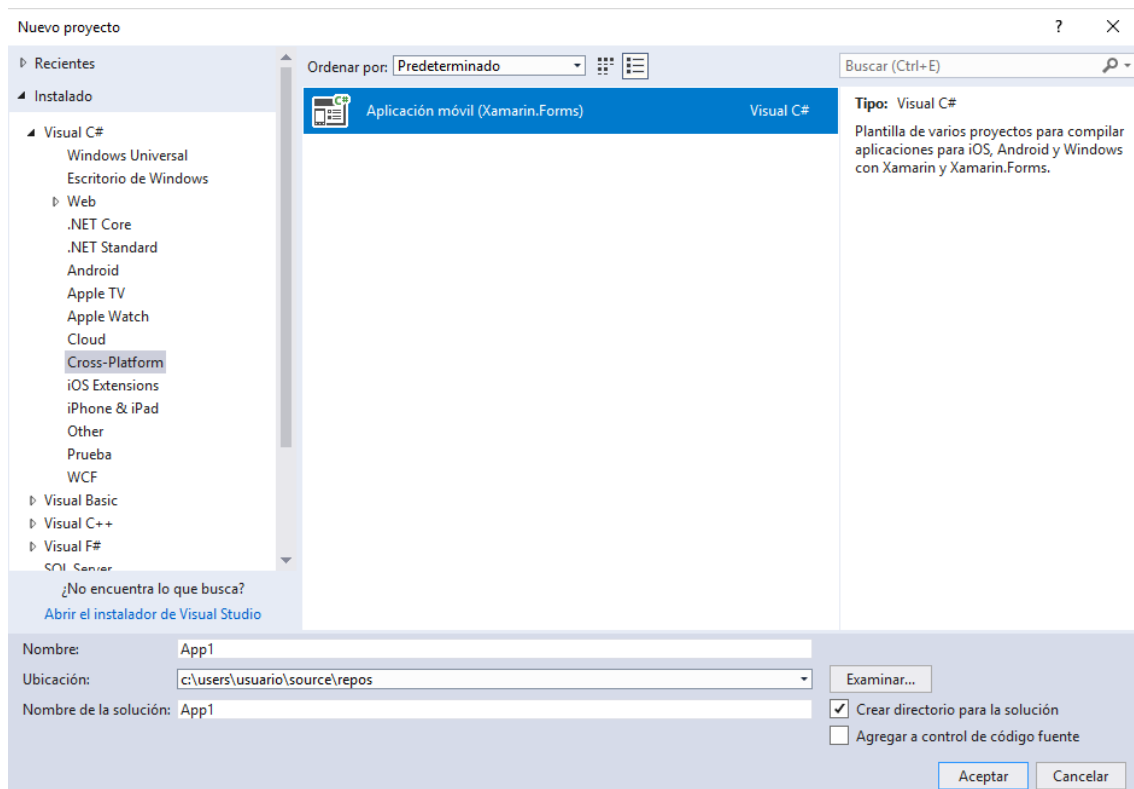


Ilustración 65: Elección de proyecto Xamarin Forms

El siguiente paso es el de la elección de tipo de aplicación móvil y plataformas para las que se desea desarrollar, en esta sección es en la que se decide si el proyecto corresponderá con las librerías PCL o se tratará de un proyecto compartido. En este caso debido al ámbito multiplataforma que se desea lograr y a los elementos complejos que forman la aplicación la mejor decisión es la de Proyecto compartido, los métodos que necesiten ser tratados con las librerías propias de cada plataforma harán uso de una compilación condicional afectando solo a una de las soluciones, por ejemplo, la reproducción de audio o el tratamiento de imágenes ya sea cambio de color o dividir una imagen en diferentes piezas.

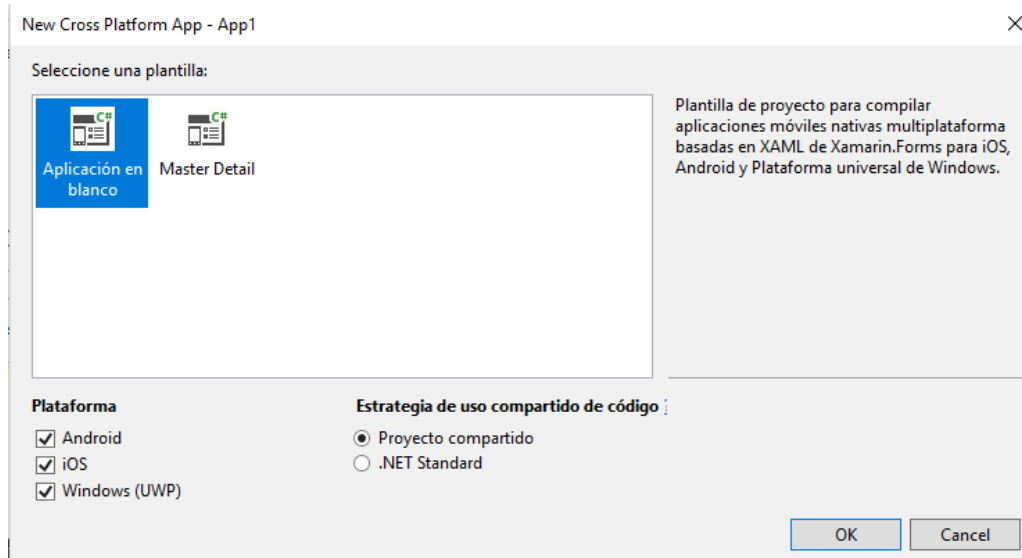


Ilustración 66: Tipo de solución Xamarin Forms

Una vez creado el proyecto se dividirá en el entorno de trabajo con los siguientes directorios:

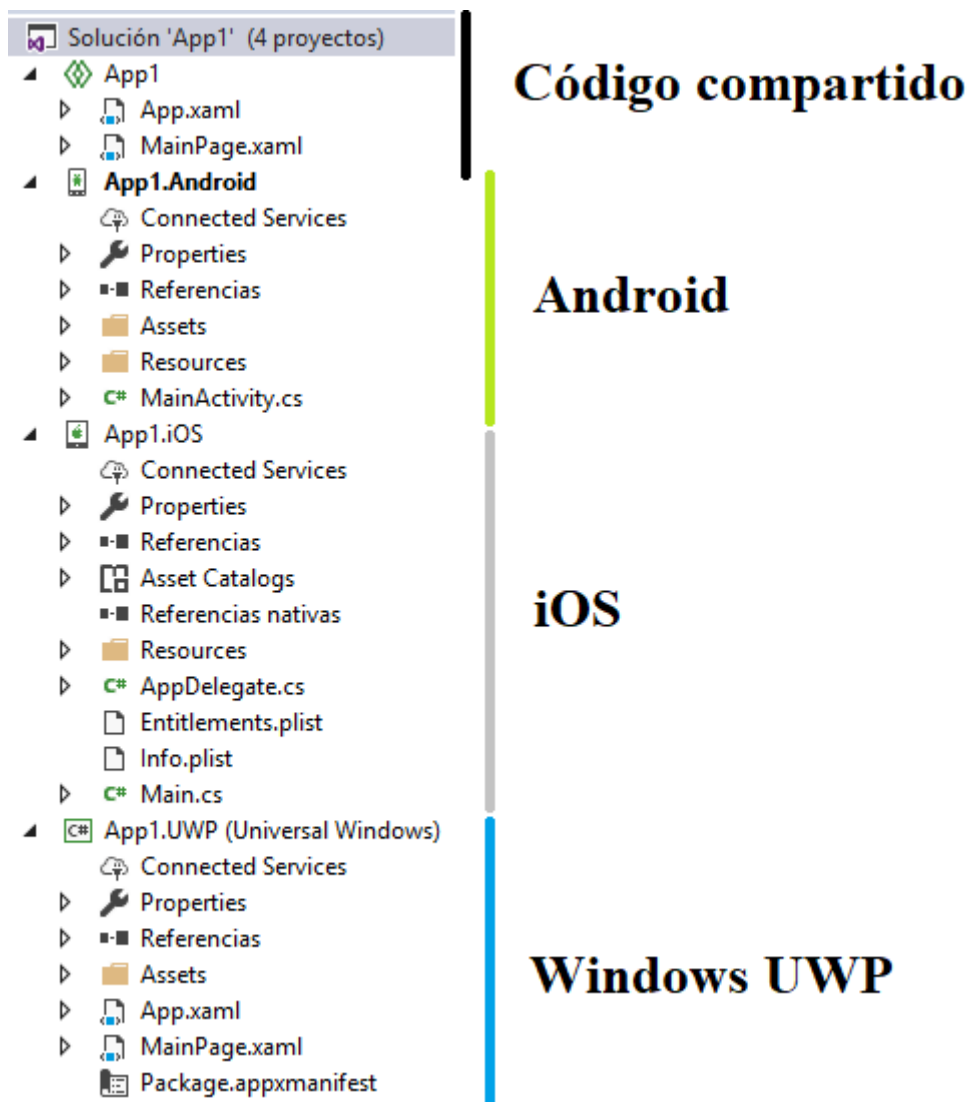


Ilustración 67: Directorios solución Xamarin Forms



Existe una zona de código compartido y otras secciones que corresponden a la aplicación en cada determinada plataforma. Todo el código importante de la aplicación se sitúa en la zona de código compartido. En los directorios de las plataformas individuales se sitúan ficheros como podrían ser imágenes que deben ser almacenadas dentro del proyecto, o clases para el manejo de las capacidades internas de cada dispositivo, por ejemplo, la obtención de la resolución de la pantalla del dispositivo utilizado.



ANEXO 2: VISUAL STUDIO APP CENTER

Una vez iniciado el desarrollo de la aplicación es importante la fase de depuración, donde se realizan diferentes pruebas en busca de errores de ejecución. Estos errores en ocasiones se pasan por alto si no se tratan de la forma adecuada. Visual Studio App Center proporciona un entorno de trabajo en el que almacenar diferentes analíticas de ejecuciones de la aplicación, diagnóstico de errores, compilación, test y distribución. Para poder beneficiarse de estas ventajas únicamente debe instalarse el *plugin* de App Center necesario. Desde el administrador de paquetes NuGet de la solución se deben instalar los siguientes complementos:

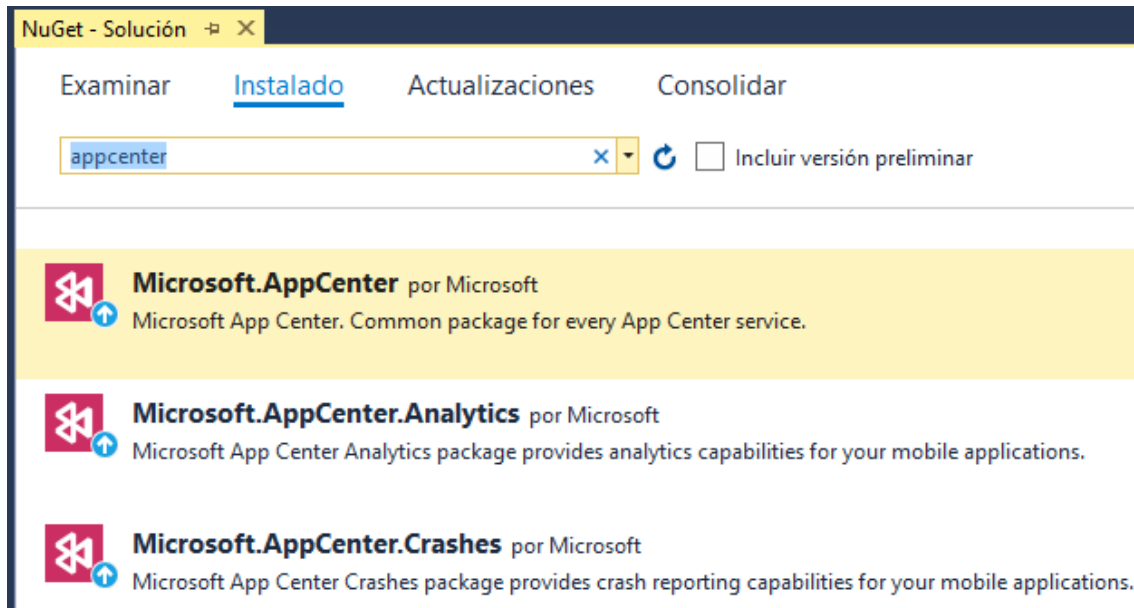


Ilustración 68: Paquetes App Center

Una vez instalados los diferentes paquetes, basta con realizar la llamada en el método `OnStart`, común para todas las plataformas, de este modo se registra la ejecución de la aplicación. De este modo es posible registrar, sesiones diarias, usuarios activos, duración de las sesiones o incluso dispositivo en el que se ejecuta la aplicación.

```
protected override void OnStart ()
{
    // Handle when your app starts
    AppCenter.Start("android=asdas23-9ea0-1232-12323-23dswfdfsdf;" +
        "uwp=213dsds-12sd-cdsx23-sdq234-sda123432r;" +
        "ios=asdas123-12ed2-123dqa-123swq-123dfsqaqderw",
        typeof(Analytics), typeof(Crashes));
}
```

Ilustración 69: Registro de analíticas App Center

No solo se dispone de ese control de ejecuciones, si no que se también hay la posibilidad de registro de excepciones, a fin de poder comprobar de forma online los errores controlados y no controlados que se produzcan.

El *plugin* almacena automáticamente las excepciones no controladas en el código, es posible añadir las excepciones controladas mediante código, este código no es soportado en la solución de Windows UWP, por lo que en Windows únicamente es posible registrar las excepciones no controladas.

```
#if __ANDROID__
    Crashes.TrackError(ex);
#elif __IOS__
    Crashes.TrackError(ex);
#endif
```

Ilustración 70: Código control excepciones controladas App Center

En caso de saltar una excepción, el método de visualizarlas es acceder a la región de diagnóstico de la plataforma que se quiera revisar. La propia web permite la notificación de excepciones a través de correo electrónico, avisando al programador o encargado de ello de que revise la web ya que se ha producido un nuevo error. Estas excepciones pueden visualizarse directamente en la web del siguiente modo:

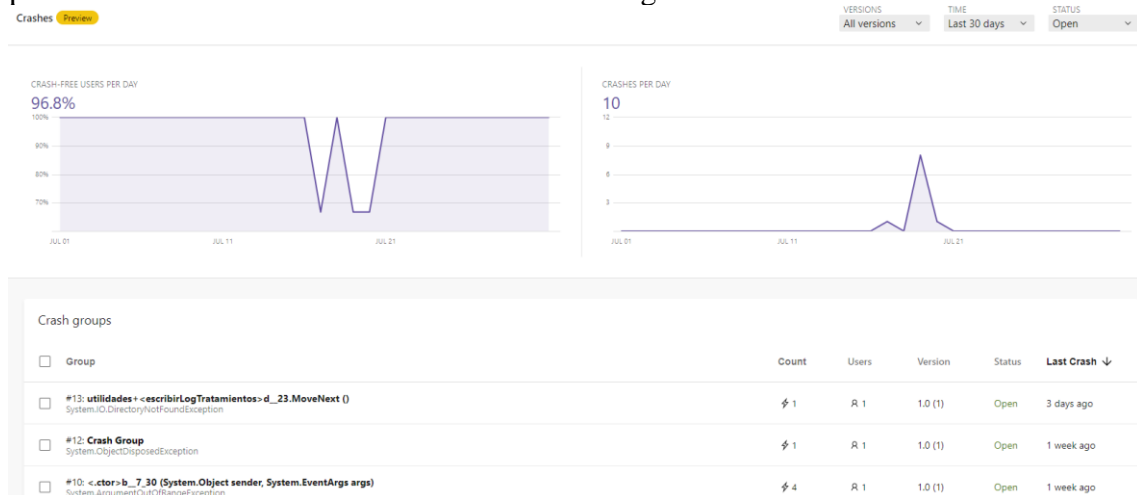


Ilustración 71: Visualización excepciones App Center

Se agrupan las excepciones y se puede realizar un control de las excepciones que saltan en determinados días, o el tiempo que llevamos sin ella, y en caso de desear ver una excepción en concreto basta con hacer clic sobre ella para ver como se ha producido.

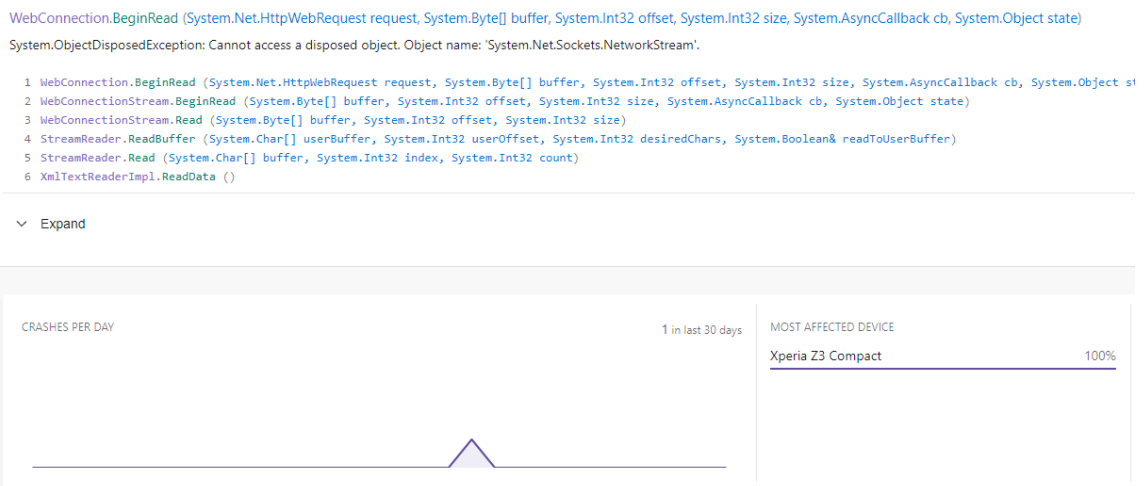


Ilustración 72: Visualización de una excepción App Center

Otras opciones que nos da esta herramienta es la de compilar, probar o incluso distribuir la aplicación, ya sea a través de la tienda de aplicaciones de cada plataforma como de forma independiente. Es posible distribuir la aplicación a diferentes grupos de usuarios seleccionados para el proceso de prueba de la aplicación, registrando la información relevante de este proceso.