

Universidad de Alcalá  
Escuela Politécnica Superior

Grado en tecnologías de la telecomunicación



ESCUELA POLITECNICA  
Autor: Antonio Sanz Corbalán  
Tutor/es: Antonio García Herraiz

2017

## Índice

|  |    |
|--|----|
| Índice de figuras .....  | 4  |
| Índice de tablas .....   | 5  |
| Resumen.....   | 6  |
| Summary .....  | 6  |
| Palabras Clave: .....  | 6  |
| Resumen Extendido .....  | 7  |
| Memoria .....  | 8  |
| Introducción .....   | 8  |
| Base teórica .....   | 9  |
| Definición de TEA:.....  | 9  |
| Prevalencia: .....   | 10 |
| Importancia de la detección precoz en los TEA .....                      | 10 |
| Diagnóstico .....  | 11 |
| Inmediatas .....   | 13 |
| Antes de los 12 meses.....   | 13 |
| Después de los 12 meses .....  | 13 |
| Etiología:.....  | 13 |
| Factores genéticos.....  | 13 |
| Instrumentos para la evaluación del autismo y síndrome de Asperger ..... | 14 |
| mHealth y aplicaciones relacionadas con TEA: .....                       | 15 |
| Aplicaciones para niños con TEA.....                                     | 17 |
| iPrompts .....   | 17 |
| Gaido.....   | 17 |
| Autism Track .....   | 18 |
| The Social Express.....  | 18 |
| Stories2Learn .....  | 19 |
| Model Me Going Places.....   | 19 |
| Cognoa .....   | 20 |
| MITA .....   | 20 |
| Conclusiones.....  | 21 |
| Análisis de datos.....   | 21 |

|   |    |
|---|----|
| Descripción del Proyecto.....                       | 24 |
| Crear un proyecto.....                              | 24 |
| AndroidManifest.xml.....                            | 26 |
| Gradle .....  | 29 |
| Activities .....                                    | 30 |
| Layouts.....  | 32 |
| Canvas.....   | 34 |
| Splash Screen.....                                  | 36 |
| onTouchEvent.....                                   | 36 |
| MotionEvent.....                                    | 37 |
| Multi-Touch .....                                   | 37 |
| Velocity Tracker.....                               | 38 |
| Animaciones .....                                   | 38 |
| MediaPlayer.....                                    | 39 |
| AlertDialog .....                                   | 39 |
| Comunicación entre base de datos y aplicación ..... | 40 |
| Base de datos.....                                  | 40 |
| Diseño página web .....                             | 47 |
| Página web .....                                    | 50 |
| Código HTML y PHP: .....                            | 51 |
| Código CSS: .....                                   | 52 |
| Manual de usuario .....                             | 54 |
| Instalación Android Studio .....                    | 54 |
| Instalación Genymotion .....                        | 54 |
| Kobiton .....                                       | 56 |
| Resumen funcional de la aplicación.....             | 57 |
| 1.    Splash Screen .....                           | 57 |
| 2.    Login .....                                   | 58 |
| 3.    Sign Up .....                                 | 59 |
| 4.    Menú .....                                    | 60 |
| 5.    Puzzles.....                                  | 61 |
| 6.    Pantalla marcadora de nivel.....              | 62 |
| 7.    Elegir puzzle .....                           | 63 |
| 8.    Opciones.....                                 | 64 |
| 9.    Selección de fondo .....                      | 64 |

|     |   |    |
|-----|---|----|
| 10. | Grabación de audio .....                    | 65 |
| 11. | Instrucciones .....                         | 66 |
|     | Diagrama de Flujo.....                      | 67 |
|     | Aplicaciones nativas vs hibridas .....      | 67 |
|     | Diagrama de Gantt .....                     | 68 |
|     | Exportar base de datos con phpMyAdmin ..... | 69 |
|     | Conclusiones y futuros trabajos .....       | 70 |
|     | Bibliografía .....                          | 71 |

# Índice de figuras

|  |    |
|--|----|
| Imagen 1. Triada de Wing .....                                       | 9  |
| Imagen 2. Adaptación e iniciativas mHealth.....                      | 16 |
| Imagen 3. Estados miembros con al menos una iniciativa mHealth ..... | 17 |
| Imagen 4. Aplicación iPromts.....                                    | 17 |
| Imagen 5. Aplicación Gaido.....                                      | 18 |
| Imagen 6. Aplicación AutismTrack .....                               | 18 |
| Imagen 7. Aplicación The social Express .....                        | 19 |
| Imagen 8. Aplicación Stories2Learn .....                             | 19 |
| Imagen 9. Aplicación Model Me Going Places .....                     | 20 |
| Imagen 10. Aplicación Cognoa .....                                   | 20 |
| Imagen 11. Aplicación MITA.....                                      | 21 |
| Imagen 12. Aplicación MITA.....                                      | 21 |
| Imagen 13. sentidos y signos Track Velocity Method.....              | 22 |
| Imagen 14. Crear un proyecto.....                                    | 24 |
| Imagen 15. Versiones Android .....                                   | 25 |
| Imagen 16. Selección dispositivo .....                               | 25 |
| Imagen 17. Estructura proyecto Android Studio .....                  | 26 |
| Imagen 18. Ciclo de vida de una Activity .....                       | 31 |
| Imagen 19. Métodos ciclo de vida Activity .....                      | 32 |
| Imagen 20. Layout con su código xml .....                            | 33 |
| Imagen 21. Splash Screen.....  | 36 |
| Imagen 22. Esquema BBDD modelo relacional.....                       | 41 |
| Imagen 23. tablas de la base de datos .....                          | 41 |
| Imagen 24. Esquema comunicación Servidor-BBDD-App .....              | 42 |
| Imagen 25. UML diagrama Singleton.....                               | 48 |
| Imagen 26. Captura antoniosanz.hol.es.....                           | 50 |
| Imagen 27. Instalación Android Studio .....                          | 54 |
| Imagen 28. Instalación Genymotion .....                              | 55 |
| Imagen 29. Simulación con Genymotion de Nexus 9.....                 | 55 |
| Imagen 30. Dispositivos Kobiton .....                                | 56 |
| Imagen 31. Aspecto controlador dispositivo Kobiton.....              | 57 |
| Imagen 32. Splash inicial .....                                      | 57 |
| Imagen 33. Pantalla de logeo .....                                   | 58 |
| Imagen 34. Mensaje de alerta pantalla login.....                     | 59 |
| Imagen 35. Pantalla de registro .....                                | 59 |
| Imagen 36. Mensaje de alerta pantalla registro .....                 | 60 |
| Imagen 37. Pantalla menú principal.....                              | 60 |
| Imagen 38. Mensaje de alerta cerrar Sesión.....                      | 61 |
| Imagen 39. Puzle 1 .....   | 61 |
| Imagen 40. Pantalla marcador nivel.....                              | 62 |
| Imagen 41. Alerta salir del juego.....                               | 63 |

|  |                                      |
|--|--------------------------------------|
| Imagen 42. Elegir puzzle .....         | 63                                   |
| Imagen 43. Menú opciones.....          | 64                                   |
| Imagen 44. Selección color fondo ..... | 64                                   |
| Imagen 45. Grabar Audio .....          | 65                                   |
| Imagen 46. Instrucciones juegos.....   | 66                                   |
| Imagen 47. Diagrama aplicación .....   | <b>¡Error! Marcador no definido.</b> |

## Índice de tablas

|  |    |
|--|----|
| Tabla 1. Estimación de prevalencia en España (1999 -2008)..... | 10 |
| Tabla 2. Tareas .....  | 68 |
| Tabla 3. Diagrama de Gantt .....                               | 69 |

## Resumen

El trabajo consiste en el desarrollo de una aplicación Android, capaz de detectar anomalías en el desarrollo de niños a temprana edad. La aplicación se compone de puzzles interactivos que evalúan la capacidad de integración de objetos teniendo en cuenta sus características (forma, color y tamaño). Un niño afectado puede mostrar dificultades relacionando características comunes entre varios objetos de una misma imagen. El objetivo es detectar casos a temprana edad con dificultades para resolver los puzzles adecuadamente y/o tratar de desarrollar la habilidad de integración en niños ya diagnosticados con algún tipo de anomalía.

## Summary

The Android application developed, is capable of an early screening child disorder. The application consists of various simple puzzles that challenges the user to focus in one or two features of the images (shape, color, size) and to solve the puzzle according to those features. Researches show that some child disorders can be detected by tracking the behavior of the child when he or she plays. Therefore the main objective of the application is to notify whenever a child's behavior is unusual and also as a breakthrough into the child learning ability.

## Palabras Clave:

TEA, autismo, detección, temprano, infantil, Android.

## Resumen Extendido

El trabajo consiste en desarrollar una aplicación para dispositivos Android que ayuda a detectar anomalías en niños a temprana edad. Esta idea se basa en diferentes estudios que demuestran la capacidad que tienen algunos juegos para ayudar a detectar ciertas anomalías. Estos métodos son utilizados actualmente por especialistas para realizar diagnósticos de algunos trastornos infantiles. Con ayuda de la tecnología hemos desarrollado una serie de juegos similares que capturan información relevante mientras el usuario juega con la aplicación. Con el software de la aplicación guardamos toda la información en variables y posteriormente las almacenamos en una base de datos donde por último, un especialista pueda determinar el diagnóstico consecuente. Registraremos variables como la velocidad de arrastre de las piezas del puzzle, los tiempos entre pulsaciones, el número de fallos y tres variables N1, N2 y N3 correspondientes al Jerk. El jerk mide los cambios de aceleración y se utiliza para determinar el número de variaciones a través de un eje fijado. En puzzles donde resulta más difícil ajustar las piezas vemos que la variable N1 (corresponde con las variaciones de tipo más estrechas) suele coincidir con valores más altos que en puzzles donde resulta más sencillo ajustar en los huecos.

Para el desarrollo de la mayor parte de la aplicación hemos utilizado el lenguaje de programación Java orientado a Android. La parte principal de la aplicación son cada uno de los puzzles, donde tenemos definido la función `onDraw` que implementamos para realizar el diseño del puzzle. Dentro de este método se define cada pieza del puzzle que será dinámica y la posición de los huecos donde deberán ir colocadas cada pieza. De esta forma al iniciar la actividad de un puzzle, lo primero que realiza el programa es pintar el diseño del puzzle correspondiente. Una vez terminado este proceso, el usuario puede arrastrar cualquier pieza alrededor de la pantalla gracias al método `onTouchEvent` que define varios los distintos gestos que puede realizar el usuario sobre la pantalla (por ejemplo `ACTION_MOVE`). Cada pieza tiene del puzzle tiene un identificador propio y un flag que se activa cuando la posición en los ejes de la pantalla es igual al de su hueco correspondiente. De esta manera podemos saber cuándo el puzzle está resuelto correctamente.

Al finalizar un puzzle correctamente pasamos a una pantalla intermedia que muestra el nivel del juego en el que nos encontramos y el número de fallos obtenido. Durante esta pantalla realizamos un push en la base de datos con todos los registros guardados en el puzzle anterior.

Los resultados que obtenemos a través de la aplicación permiten realizar un primer reconocimiento del comportamiento del usuario y comprobar si se aleja de los resultados estándares o presenta unos resultados distintos. En este último caso, un especialista en diagnósticos de trastornos infantiles podría realizar las pruebas adecuadas para determinar si existe alguna anomalía.

Siguiendo este protocolo, con el uso aplicaciones mHealth se conseguir detectar anomalías con mayor rapidez y de esta manera empezar con el tratamiento más temprano, lo cual es muy importante en muchos de los trastornos infantiles.

# Memoria

## Introducción

El concepto de “Mental Imagery Therapy for Autism” (MITA) es un sistema de detección de trastornos de espectro autista (TEA) basado en un tratamiento de integración de objetos mediante puzzles sencillos. MITA se basa en “Pivotal Response Treatment” (PRT), una de las mejores terapias para niños diagnosticados con Autismo. El objetivo de MITA es entrenar la habilidad de diferenciar entre varias características presentadas simultáneamente sobre una misma imagen. Esta habilidad está deteriorada en personas que sufren TEA y se conoce comúnmente como visión túnel [3].

Actualmente la media de edad con la que se detecta TEA es de seis años. El Centro de prevención de enfermedades (CDC) considera necesario reducir la media edad de su detección a los tres años. [1].

Testea, se basa en el concepto MITA que mejora habilidades como el lenguaje, la atención y la percepción visual. Entrenando estas capacidades se puede lograr que un niño/a con TEA sea capaz de centrarse en una o múltiples características de un objeto y en el conjunto global de objetos que presenta una imagen, lo cual supone un gran avance en su aprendizaje [2].

Actualmente los psicólogos y especialistas de anomalías en el desarrollo infantil utilizan tests y pruebas psicotécnicas para diagnosticar a los pacientes. Estas pruebas son gracias a la tecnología actual reproducibles mediante aplicaciones móviles, y además pueden ayudar a reducir los costes y la edad media de detección que supone un factor decisivo en la calidad de vida del paciente.

Para analizar el comportamiento de niños, hemos desarrollado una aplicación capaz de registrar una serie de variables mientras el usuario utiliza la aplicación. Se trata de una aplicación nativa desarrollada en Android que contiene una serie de puzzles con fichas de distintas formas, colores e incluso imágenes que el usuario deberá colocar en la posición correcta. Mientras el usuario juega, la aplicación hace un registro en segundo plano guardando variables como la velocidad y aceleración de los movimientos con el dedo sobre la pantalla o de los intervalos de tiempo entre pulsaciones. Las variables se almacenan en una base de datos y quedan ordenados según el nombre de usuario y el número de puzzle realizado.

El objetivo de la aplicación es detectar indicios o patrones característicos de niños con enfermedades de espectro autista o enfermedades desarrolladas a edad temprana. De esta forma gracias a este tipo de aplicaciones, sería posible detectar antes la enfermedad y comenzar a una edad más temprana con el tratamiento y/o ejercicios de aprendizaje.

# Base teórica

## Definición de TEA:

Los TEA se definen como un conjunto de trastornos neurológicos crónicos con una fuerte base genética que desde edades tempranas se manifiesta en una serie de síntomas basados en la interacción social, comunicación y falta de flexibilidad en el razonamiento y comportamientos [3].



Imagen 1. Triada de Wing

Los TEA manifiestan síntomas muy variados y con grados distintos de gravedad, forma y edad de aparición. Cada persona con TEA presenta un cuadro de características observables distintas, por ello nos referimos a TEA como un conjunto o espectro de muchos trastornos neurológicos del desarrollo [4].

Las enfermedades de espectro autista (TEA) son una gama de trastornos con características similares entre sí. Los tres tipos diferentes de TEA son:

- Trastorno autista. Es la condición más común entre los TEA y se caracteriza por grandes retrasos en el lenguaje, dificultades con las interacciones sociales y comportamientos inusuales.
- Síndrome de Asperger. Las personas con este síndrome presentan algunos de los síntomas más leves del autismo, como desafíos sociales y comportamientos inusuales. No tienen retrasos en el lenguaje ni capacidades intelectuales deficientes.
- Trastorno generalizado del desarrollo no especificado. También conocido como autismo atípico, se les puede diagnosticar si cumplen con algunos de los criterios del trastorno autista o del síndrome de Asperger.

## Prevalencia:

Tradicionalmente se ha considerado al autismo como un trastorno de escasa prevalencia (4-5 por cada 10.000 en los años 70). En la actualidad las cifras han aumentado, hasta 60 por 10.000 teniendo en cuenta todo el espectro autista.

Según la Asociación Internacional Autismo-Europa (2000), a nivel mundial, cinco de cada 10.000 personas presentan un cuadro de “autismo clásico”, y las cifras ascienden hasta 700-1000 personas si se tiene en cuenta el “espectro autista” [5].

En España se han realizado pocos estudios, sin embargo las cifras estiman de 8 por cada 10.000 para el grupo de trastorno autista y Asperger entre 0-4 años y 3,5 entre 5-9 años. Según la encuesta del INE sobre discapacidades, deficiencias y estado de salud de 1999, existen en nuestro país 3,8 millones de personas con discapacidad de las cuales unas 4500 personas están diagnosticadas con autismo (0,12%) [6].

Respecto al género, se estima que los hombres tienen 4 veces más riesgos de padecer este trastorno que las mujeres. El último informe del Centro para el control y Prevención de Enfermedades (CDC, 2009) mantiene que la prevalencia de Trastornos del Espectro Autista (TEA) es significativamente mayor ( $p < 0,001$ ) en niños que en niñas [5].

Por otro lado, cabe destacar que las mujeres con autismo y sin otro tipo de déficit añadido, son menos diagnosticadas que los hombres ya que son capaces de compensar mejor las dificultades.

La edad de inicio de los trastornos es muy temprana, en algunos casos es posible detectarla en el primer año de edad, de ahí la importancia de realizar un seguimiento exhaustivo del desarrollo infantil.

| ESPAÑA              |            |
|---------------------|------------|
| POBLACIÓN TOTAL     | 46.745.807 |
| POBLACION TEA 1999  | 5224       |
| ESTIMACIÓN / 10.000 | 1.1        |
| POBLACIÓN TEA *2008 | 13.800     |
| POLACIÓN TEA **2008 | 1900       |
| ESTIMACIÓN / 10.000 | 3.35       |

Tabla 1. Estimación de prevalencia en España (1999 -2008)

\*Personas con 6 años o más de edad.

\*\*Personas con 5 años o menos de edad.

## Importancia de la detección precoz en los TEA

Detectar de manera precoz un trastorno de espectro autista desemboca en una intervención temprana en los niños y en sus familias, lo cual ayuda a reducir el estrés y a desarrollar aptitudes

sociales en la persona afectada de cara al futuro. Las instituciones de apoyo a los TEA y los profesionales afirman que una detección precoz es importante.

#### **Ventajas de la detección precoz de los TEA:**

- Previenen o rectifican diagnósticos erróneos.
- Reduce incertidumbres de los padres.
- Facilita el contacto con otras personas o familiares con TEA.
- Ayuda a la identificación de opciones de educación.
- Proporciona acceso a recursos, apoyo y servicios.
- Ayuda a identificar necesidades de apoyo familiar.

De cara al futuro, existen indicios de que la intervención temprana parece mejorar el pronóstico global de los niños con TEA. Características como la comunicación y la interacción social mejoran considerablemente si la intervención es precoz y el apoyo por parte de especialistas es el adecuado. Otro aspecto importante a favor de la detección precoz, es la raíz genética que tienen los TEA.

## **Diagnóstico**

Los criterios diagnósticos de TEA [3, 7,8] son los siguientes:

A. Déficits persistentes en comunicación e interacción social:

1. Déficits en reciprocidad socio-emocional. Problemas para mantener el flujo de ida y vuelta normal de las conversaciones. Disposición reducida para compartir intereses, emociones y afecto.
2. Déficits en conductas comunicativas no verbales usadas en la interacción social. Rango de comportamientos que van desde anomalías en el contacto visual y en el lenguaje corporal, hasta una falta de expresividad emocional o de comunicación no verbal.
3. Déficits para desarrollar, mantener y comprender relaciones. Comportamientos como dificultad para encajar en diferentes contextos sociales o dificultades para compartir juegos y hacer amigos. Hasta el punto de carencia de interés por la gente.

B. Patrones repetitivos y restringidos de conductas, actividades e intereses:

1. Movimientos motores, uso de objetos o habla estereotipados o repetitivos.
  2. Insistencia en la igualdad, adherencia inflexible a rutinas o patrones de comportamiento verbal y no verbal ritualizado. Malestar ante pequeños cambios, patrones de pensamiento rígidos y/o necesidad de seguir siempre el mismo camino.
  3. Intereses altamente restringidos, obsesivos que son anormales por su intensidad.
  4. Hiper-reactividad sensorial o interés inusual en aspectos sensoriales del entorno. Por ejemplo indiferencia aparente a dolor o temperatura, respuesta adversa a sonidos o texturas, oler o tocar objetos en exceso. Fascinación por las luces u objeto que giran.
- C. Los síntomas deben estar presentes en el periodo de desarrollo temprano. Sin embargo pueden no manifestarse plenamente hasta que las demandas del entorno exceden las capacidades del niño, o pueden verse enmascaradas en momentos posteriores de la vida.
- D. Los síntomas causan alteraciones clínicamente significativas a nivel social, ocupacional o en otras áreas importantes del funcionamiento actual.
- E. Estas alteraciones no se explican mejor por la presencia de una discapacidad intelectual o un retraso global del desarrollo. La discapacidad intelectual y el trastorno de espectro autista a menudo concurren.
- F. Algunos niños pueden mostrar habilidades especiales en áreas concretas sensoriales, de memoria, cálculo u oído.

## Señales de alerta

### Inmediatas

- No balbucea, no hace gestos (señalar, decir adiós con la mano) a los 12 meses.
- No dice palabras sencillas a los 18 meses.
- No dice frases espontáneas de dos palabras a los 24 meses.
- Cualquier pérdida de habilidades del lenguaje a o nivel social a cualquier edad.

### Antes de los 12 meses

- Poca frecuencia del uso de la mirada dirigida a personas.
- Falta de sonrisa social.
- Falta de interés en juegos interactivos simples.

### Después de los 12 meses

- No responde a su nombre
- No mira hacia donde otros señalan.
- Entonación anormal.
- Consigue cosas por sí mismo, sin pedirlas.

## Etiología:

El autismo es un trastorno cerebral que afecta a la coordinación, sincronización e integración entre las diferentes áreas cerebrales. Las alteraciones más significativas de estos trastornos son las interacciones sociales, la comunicación y las conductas repetitivas. Por ello el concepto de autismo engloba un conjunto de trastornos con distintos fenotipos conductuales y diferentes grados de intensidad en las personas que lo presentan. Los estudios más recientes indican la importancia de factores genéticos y de algunos factores ambientales que dan lugar a alteraciones cerebrales muy tempranas. Los TEA tiene por tanto una heterogeneidad etiológica primaria en la que además pueden influir factores medioambientales en los primeros meses del embarazo [3].

## Factores genéticos

El autismo es un trastorno poligénico en el que la interacción entre varios genes puede dar lugar al fenotipo característico de los TEA en grados de intensidad variable. Los resultados obtenidos apoyan la hipótesis de que una persona debe heredar entre 15 y 20 genes que interactúan de manera sinérgica para expresar el fenotipo completo del autismo. La hipótesis sostiene que cada uno de los genes que intervienen debe aportar una pequeña cantidad de riesgo para el trastorno y solo cuando esa cantidad supera un umbral, la persona presenta el fenotipo completo. Esto explica el extenso espectro fenotípico que muestran los TEA. La situación se complica más, si se tiene en cuenta la influencia ambiental en la expresividad genética [3].

La tasa de riesgo de recurrencia en familias con algún miembro con TEA es del 8%, entre 50 y 75 veces el riesgo de la población general. En familias con más de un miembro el riesgo puede aumentar [3].

El proyecto MSSNG (named for the “missing” information about Autism) lanzado en el año 2014 por la organización Autism Speaks, en colaboración con Google y el Hospital for Sick Children in Toronto analizan el ADN de 10.000 personas con autismo y sus familias. MSSNG tiene como objetivo descifrar los 3200 millones de genes que componen el genoma humano y encontrar una explicación a las causas genéticas que componen las enfermedades de espectro autista [4].

## Instrumentos para la evaluación del autismo y síndrome de Asperger

Evaluaciones rápidas del autismo:

Checklist for Autism in toddlers (CHAT). CHAT es una prueba de detección temprana de problemas de espectro autista y está dirigida a detectar indicadores psicológicos de riesgo de autismo en niños de 18 meses. Su aplicación es muy rápida (entre 20 y 30 minutos). Consta de dos apartados diferentes con un total de 14 ítems;

- En el primero se pregunta a los padres como si su hijo/a se interesa por otros niños, si utiliza gestos para señalar o interactuar con ellos de manera apropiada.
- En el segundo aparato, el propio médico presenta varias situaciones al niño relacionado con el lenguaje y el juego imaginativo para comprobar sus reacciones.

En función de la cantidad de ítems que no supera correctamente, se establece un grado de riesgo de sufrir autismo. Por tanto, es una prueba de detección temprana que alerta sobre potenciales problemas, que deben confirmarse con la administración de otras pruebas más completas y con el seguimiento en el tiempo.

La otra prueba de detección rápida es un cuestionario de comunicación social (SCQ). Es un instrumento de detección temprana destinado a evaluar de forma rápida las capacidades de comunicación y de relación social de niños que pudiesen padecer TEA. Permite recoger los síntomas observados por los padres para poder decidir adecuadamente si es conveniente remitirlos a una evaluación más profunda.

La evaluación extensa del autismo:

Entrevista diagnóstica de autismo (ADI-R). ADI-R es una entrevista semiestructurada que permite la evaluación detallada de aquellos niños y adultos que muestran un posible trastorno del autismo. Es un instrumento muy útil para el diagnóstico formal de niños a partir de 2 años y su aplicación lleva unas dos horas.

Está compuesto por 93 ítems que se centran en tres aspectos o dominios principales: lenguaje y comunicación, interacciones sociales recíprocas, y comportamientos estereotipados y repetitivos. De esta forma, se cubren un total de ocho áreas de contenido relacionado con el autismo:

- Historia del sujeto.

- Desarrollo temprano.
- Adquisición del lenguaje y pérdida del lenguaje.
- Funcionamiento actual en los aspectos lingüísticos y comunicativos.
- Desarrollo social y del juego.
- Intereses y comportamiento.
- Comportamientos clínicamente relevantes.

Los resultados del ADI-R señalan que es bastante efectivo a la hora de distinguir entre problemas de autismo y otros cuadros similares.

Childhood Autism Rating Scale (CARS). CARS es una de las escalas más empleadas y fiables para identificar a niños con autismo y a distinguirlos de los niños con dificultades del desarrollo. Es un instrumento dirigido a niños de 2 años y compuesto por 15 ítems. Cada uno de estos ítems emplea una escala de siete puntos, de manera que el evaluador indica el grado en que el niño evaluado se aleja para cada característica concreta de los que cabría esperar de un niño normal. Una vez que se ha obtenido una puntuación total aplicando los resultados de todos los ítems, los niños cuya puntuación se sitúe sobre un punto de corte, se pueden considerar autistas.

#### Autism Diagnostic Observation Schedule (ADOS)

Es una escala semi-observacional que puede emplearse para evaluar posibles casos de autismo en pacientes de todas las edades y competencias lingüísticas. Para ello ADOS plantea diversas actividades que generan un contexto controlado que permite al evaluador observar los comportamientos sociales y comunicativos relacionados con el autismo.

Está compuesta por cuatro módulos de actividades, de forma que cada persona evaluada sólo pasa por uno de ellos dependiendo de su edad y sus capacidades lingüísticas.

- Módulo 1: diseñado para niños que no dominan la etapa de primeras frases.
- Módulo 2: diseñado para niños que ya dominan las primeras frases, pero que todavía no tienen una fluidez verbal completamente desarrollada.
- Módulo 3: diseñado para niños que tengan el lenguaje completamente adquirido.
- Módulo 4: diseñado para adolescentes y adultos.

Entre las actividades planteadas dentro de estos módulos para servir como contextos para la observación, aparecen el juego libre, tareas de construcción con bloques, responder a nombres, imitar una tarea después de ver un modelo, describir imágenes y/o contar historias escritas [7].

#### mHealth y aplicaciones relacionadas con TEA:

El Global Observatory for eHealth (GOE) define mHealth o mobile Health como la práctica médica realizada por dispositivos portables como, móviles, Tabletas o dispositivos monotorizadores. Las aplicaciones mHealth requieren el uso y la capitalización de funcionalidades como el servicio de mensajes cortos (SMS), tercera y cuarta generación de telecomunicaciones móviles (sistemas 3G y 4G), sistema de posicionamiento global (GPS) y/o Bluetooth.

Un estudio llevado a cabo por el GOE ha reconocido que las iniciativas más comunes son telephone help lines (59%), emergency toll-free telephone services (55%) y mobile telemedicine (49%). Las iniciativas mHealth menos comunes son awareness raising (23%), surveillance (26%) y Decision support systems (19%).

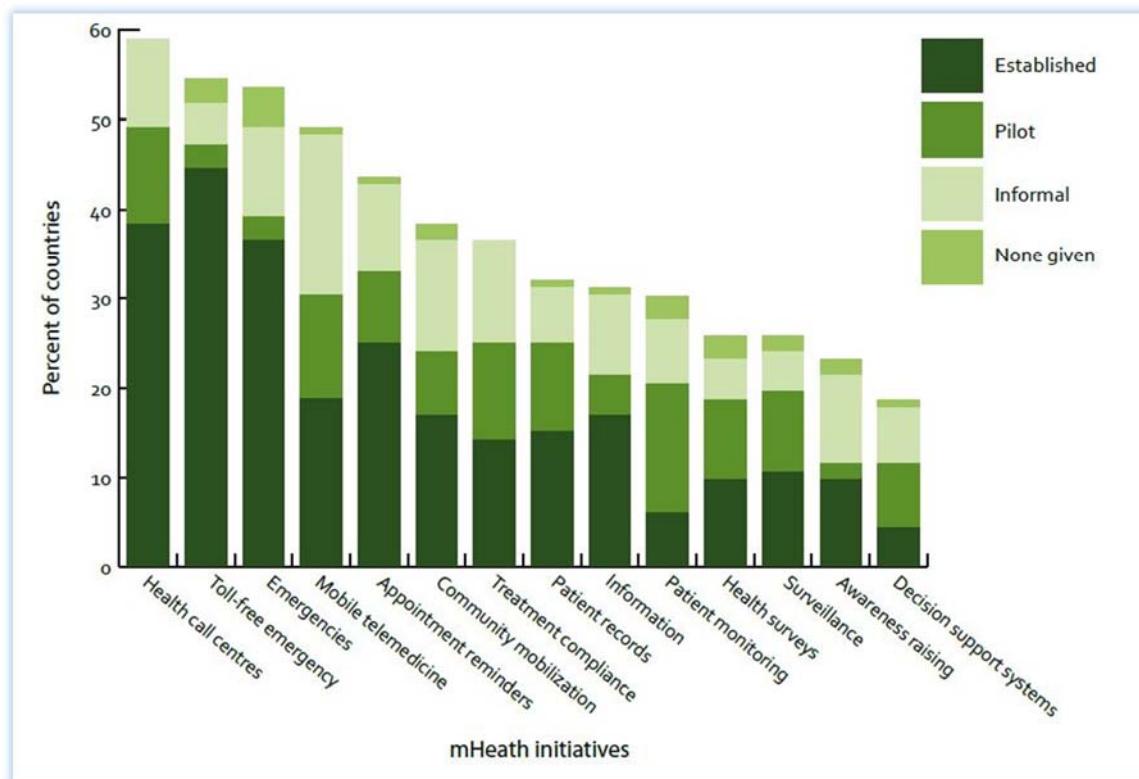
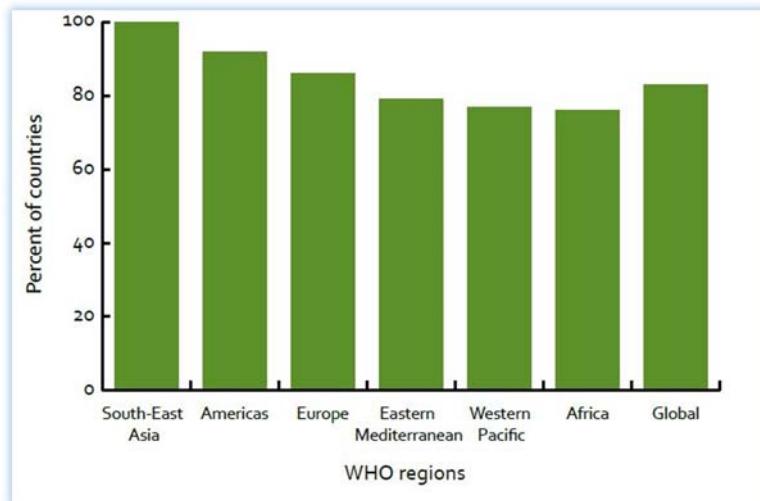


Imagen 2. Adaptación e iniciativas mHealth

El número de iniciativas mHealth en países desarrollados es mayor que en países menos desarrollados debido a la implantación de redes de comunicaciones y al acceso de dispositivos móviles en la población.

Actualmente las aplicaciones eHealth están en auge debido a la facilidad y comodidad que ofrecen al usuario. Los dispositivos móviles ofrecen la posibilidad de realizar una presunción diagnóstica a distancia, siendo necesario una confirmación por parte del personal sanitario especializado [8].



*Imagen 3. Estados miembros con al menos una iniciativa mHealth*

## Aplicaciones para niños con TEA

### iPrompts

iPrompts es una aplicación que permite crear horarios y orden de tareas visuales, utilizando imágenes, temporizadores y sistemas de elección. iPrompts ofrece una posibilidad de comunicación visual con niños afectados por TEA, sencilla y rápida [9].



*Imagen 4. Aplicación iPromts*

### Gaido

Es una plataforma software que permite la cooperación entre familiares y profesionales para ayudar a personas con TEA en su desarrollo y aprendizaje. La aplicación consiste en una serie de pictogramas que permiten la comunicación y programación de actividades entre los afectados por TEA y sus tutores, ya sean padres o expertos en TEA [10].

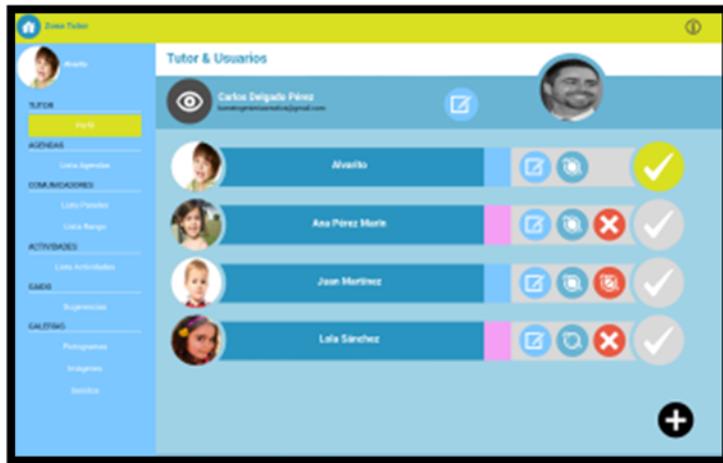


Imagen 5. Aplicación Gaido

## Autism Track

Autism Track es una herramienta para registrar información que sirve de ayuda a padres y/o tutores controlar comportamientos y síntomas de un niño con TEA. Mediante “checkboxes” y “sliders”, Autism Track registra las dietas, medicamentos, síntomas y comportamientos a lo largo del día. De esta forma Autism Track puede ayudar a los padres o tutores a determinar que terapias están funcionando mejor y cuales peor [11].



Imagen 6. Aplicación AutismTrack

## The Social Express

The Social Express es un programa online cuya función principal es mejorar las habilidades sociales en niños con problemas de aprendizaje. El programa utiliza presentaciones visuales que son capaces de interactuar con los niños y mejorar de una forma sencilla y divertida sus habilidades sociales [12].



Imagen 7. Aplicación The social Express

## Stories2Learn

Stories2Learn permite crear historias con fotos, texto y audios para promover la comunicación no verbal entre los usuarios y además lo hace de una forma entretenida y sencilla [13].

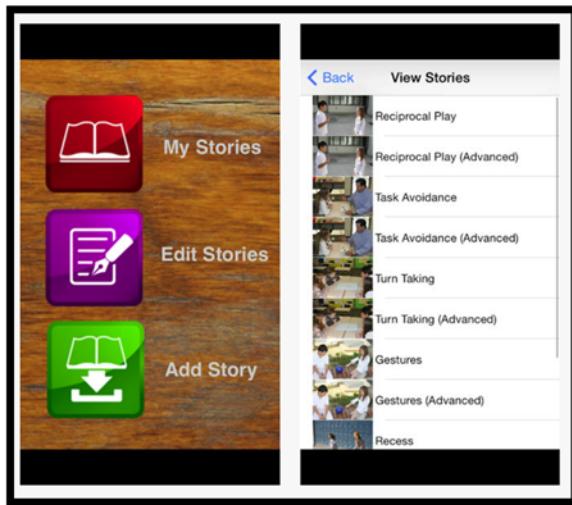


Imagen 8. Aplicación Stories2Learn

## Model Me Going Places

Model Me Going Places es una página web que contiene con una serie de videos que muestran niños realizando situaciones en lugares cotidianas como: El colegio, el parque, la peluquería, el supermercado.... Esta aplicación pretende desarrollar las habilidades de comportamiento en niños con dificultades en este aspecto [14].



Imagen 9. Aplicación Model Me Going Places

## Cognoa

La doctora Clara Lajonchere (Vicepresidenta del programa clínico Autism Speaks) ha desarrollado una herramienta de evaluación que mediante una serie de preguntas y respuestas a los padres o tutores del niño, es capaz de detectar si hay riesgo de que tenga algún tipo de trastorno de espectro autista [15].

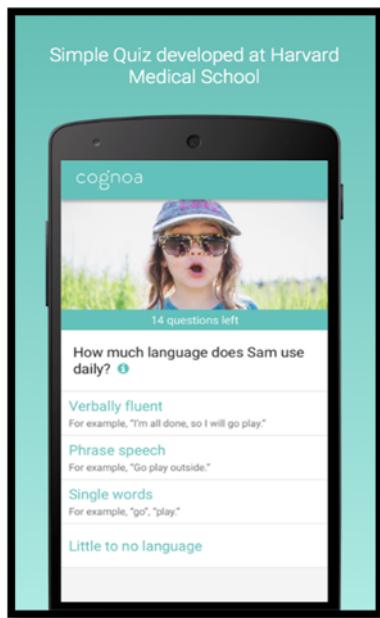


Imagen 10. Aplicación Cognoa

## MITA

Mental Imagery Therapy for Autism es una aplicación desarrollada para ayudar a niños con TEA en edades tempranas. MITA contiene una gran variedad de puzzles interactivos diseñados para ayudar a los niños a mejorar sus capacidades mentales de integración de objetos. Esta habilidad ha demostrado ser importante en el aprendizaje de niños con TEA [16].



Imagen 11. Aplicación MITA

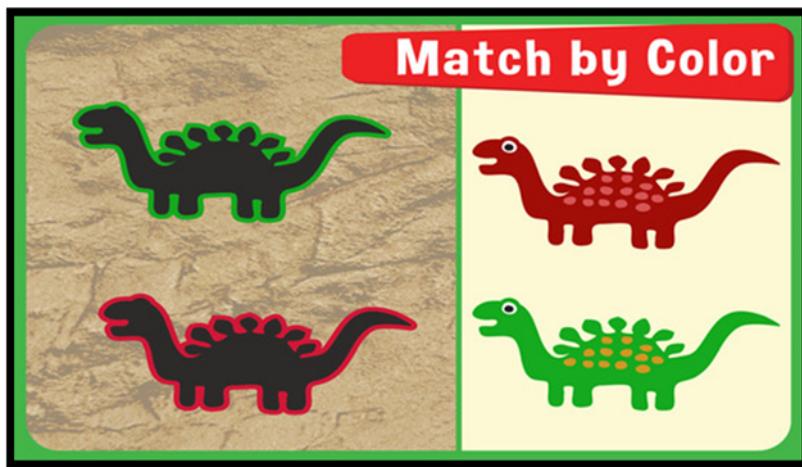


Imagen 12. Aplicación MITA

## Conclusiones

Después del análisis realizado sobre la variedad de aplicaciones y páginas web relacionadas con los trastornos de espectro autista, llegamos a la conclusión de que la mayoría de ellas están dirigidas a mejorar la comunicación entre padres y niños/as afectados y su desarrollo de aprendizaje. Sin embargo no existen aplicaciones dirigidas a la detección temprana como es el caso de nuestra aplicación.

## Análisis de datos

Hemos definido tres niveles para analizar el grado de Jerk. El nivel N1 refleja aceleraciones suaves, el nivel N2 aceleraciones medias y el nivel N3 aceleraciones bruscas. De esta forma podemos identificar el tipo de gestos que realiza el usuario al jugar con la aplicación y al mover las fichas a distintas posiciones de la pantalla. Con la media del tiempo entre pulsaciones podemos conocer si el usuario es capaz de arrastrar fichas a posiciones alejadas o realiza arrastres en pequeños pasos. El número de fallos y la duración total en resolver un puzzle nos

ayudan a registrar un historial de usuario para analizar su capacidad de aprendizaje y su nivel de mejora.

La derivada de la aceleración o jerk, es una medida que representa el cambio de aceleración [1].

El jerk se define como la tasa temporal de variación de la aceleración y representa un exceso de aceleración que experimenta un cuerpo cuando se encuentra en movimiento. Como se muestra en la ecuación siguiente, el jerk puede ser definido como la primera derivada de la aceleración respecto al tiempo o como la segunda derivada de la velocidad.

$$\vec{J} = \frac{\partial \vec{a}}{\partial t} = \frac{\partial^2 \vec{v}}{\partial t}$$

En la aplicación hemos implementado este algoritmo para detectar anomalías en el comportamiento del usuario con la aplicación.

Android define dos métodos para capturar la velocidad del trazado en los ejes X e Y del usuario que son **getXVelocity** y **getYVelocity**. Estas funciones nos devuelven la velocidad de trazado del dedo en pixeles por segundo, pudiendo ser un valor negativo dependiendo de la dirección [17].



Imagen 13. Sentidos y signos Método Track Velocity

Dependiendo del nivel de aceleración que el usuario imprime sobre la pantalla se aumenta una de las tres variables definidas en la aplicación (N1, N2 y N3) correspondientes a cada nivel de Jerk. De modo que un valor alto en la variable N3 se corresponde con un número alto de gestos acelerados por parte del usuario.

```
private void jerk_function(float x_vel, float y_vel){  
    if(Math.abs(x_vel) < 150.0f && Math.abs(y_vel) < 150.0f) {  
        N1 += 1;  
    }  
  
    if(Math.abs(x_vel) > 160.0f && Math.abs(x_vel) < 400.0f &&  
    Math.abs(y_vel) > 160.0f && Math.abs(y_vel) < 400.0f) {  
        N2 += 1;  
    }  
  
    if(Math.abs(x_vel) > 410.0f && Math.abs(x_vel) < 1400.0f &&  
    Math.abs(y_vel) > 410.0f && Math.abs(y_vel) < 1400.0f) {  
        N3 += 1;  
    }  
}
```

# Descripción del Proyecto

## Crear un proyecto

Para crear un proyecto con Android Studio pinchamos en File > New > New Project, a continuación escribimos el nombre de nuestra aplicación y seleccionamos el soporte sobre el cual queremos diseñar nuestra aplicación (teléfono, tableta, televisión, reloj o gafas). Para cada uno de los dispositivos podemos seleccionar también el nivel de API (Application programming Interface).

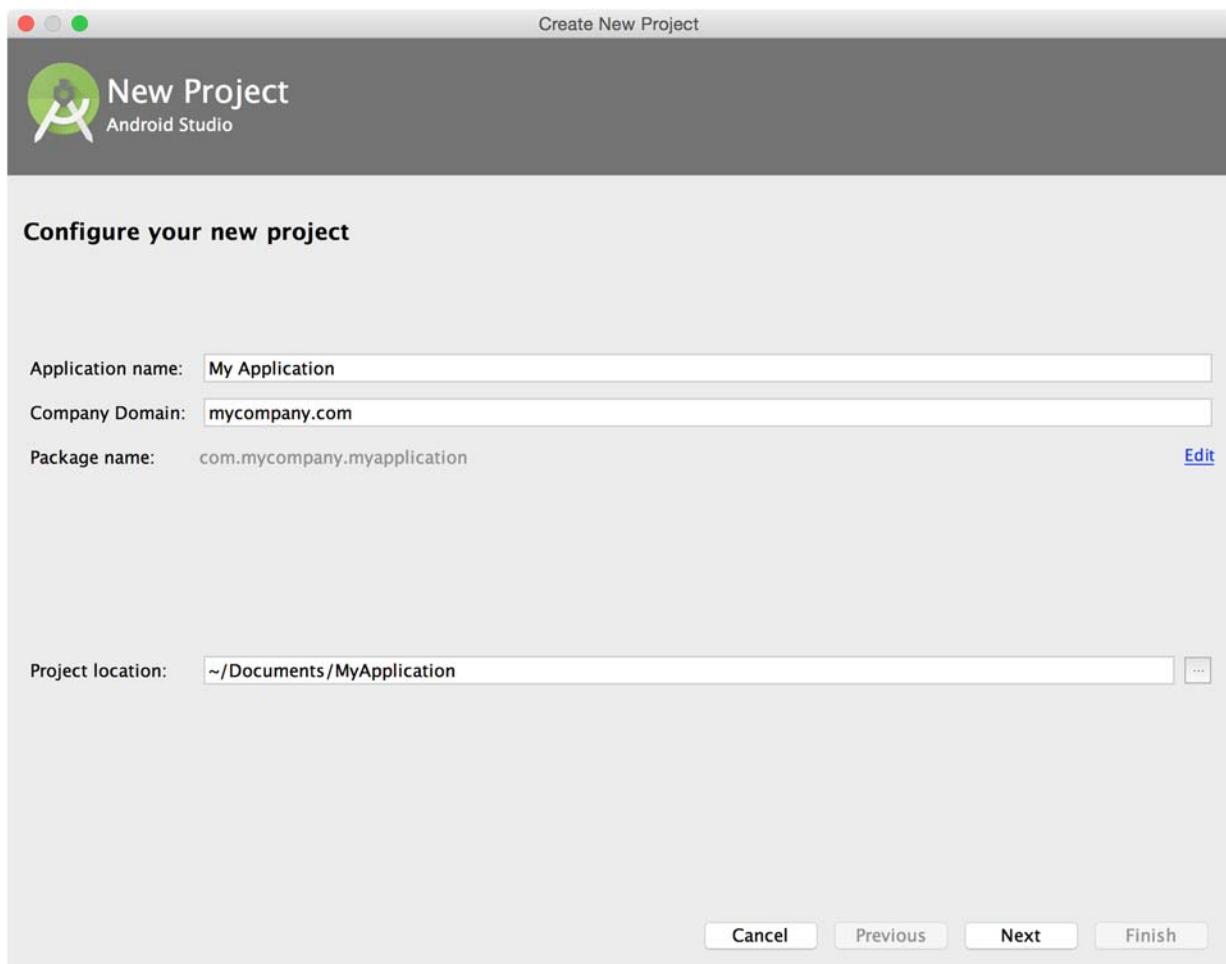


Imagen 14. Crear un proyecto

| Version       | Codename           | API | Distribution |
|---------------|--------------------|-----|--------------|
| 2.2           | Froyo              | 8   | 0.1%         |
| 2.3.3 - 2.3.7 | Gingerbread        | 10  | 1.7%         |
| 4.0.3 - 4.0.4 | Ice Cream Sandwich | 15  | 1.6%         |
| 4.1.x         | Jelly Bean         | 16  | 6.0%         |
| 4.2.x         |                    | 17  | 8.3%         |
| 4.3           |                    | 18  | 2.4%         |
| 4.4           | KitKat             | 19  | 29.2%        |
| 5.0           | Lollipop           | 21  | 14.1%        |
| 5.1           |                    | 22  | 21.4%        |
| 6.0           | Marshmallow        | 23  | 15.2%        |

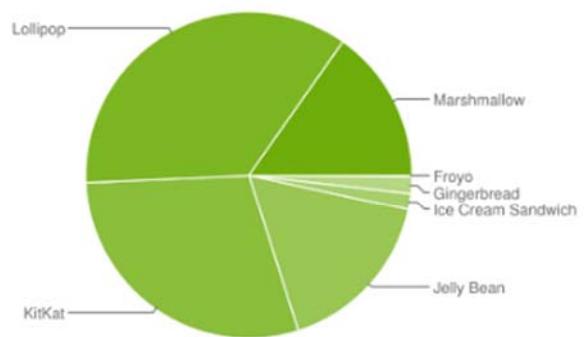


Imagen 15. Versiones Android

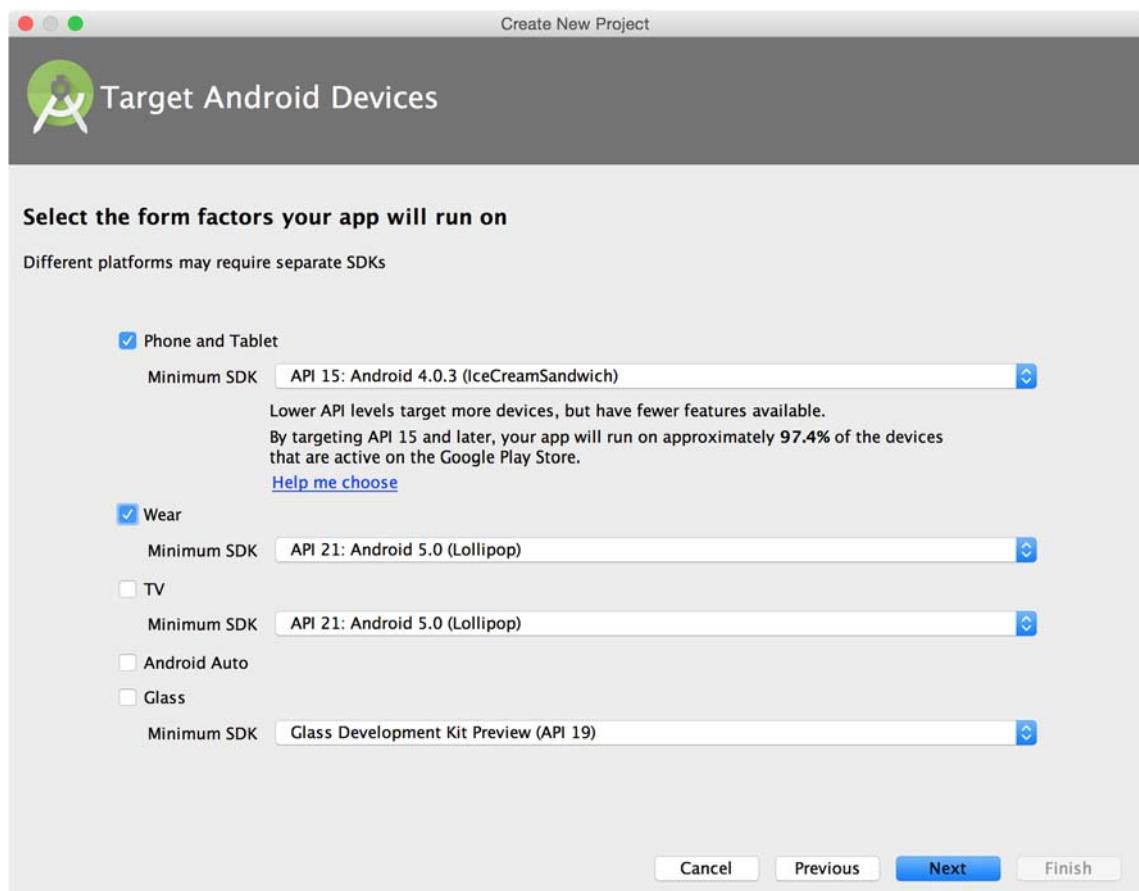


Imagen 16. Selección dispositivo

Android Studio crea por defecto la estructura definida para el proyecto. Incluye una carpeta manifests con el archivo AndroidManifest.xml, una carpeta java con las clases, y una carpeta res con las imágenes, los layouts, los sonidos y las animaciones de nuestro proyecto.

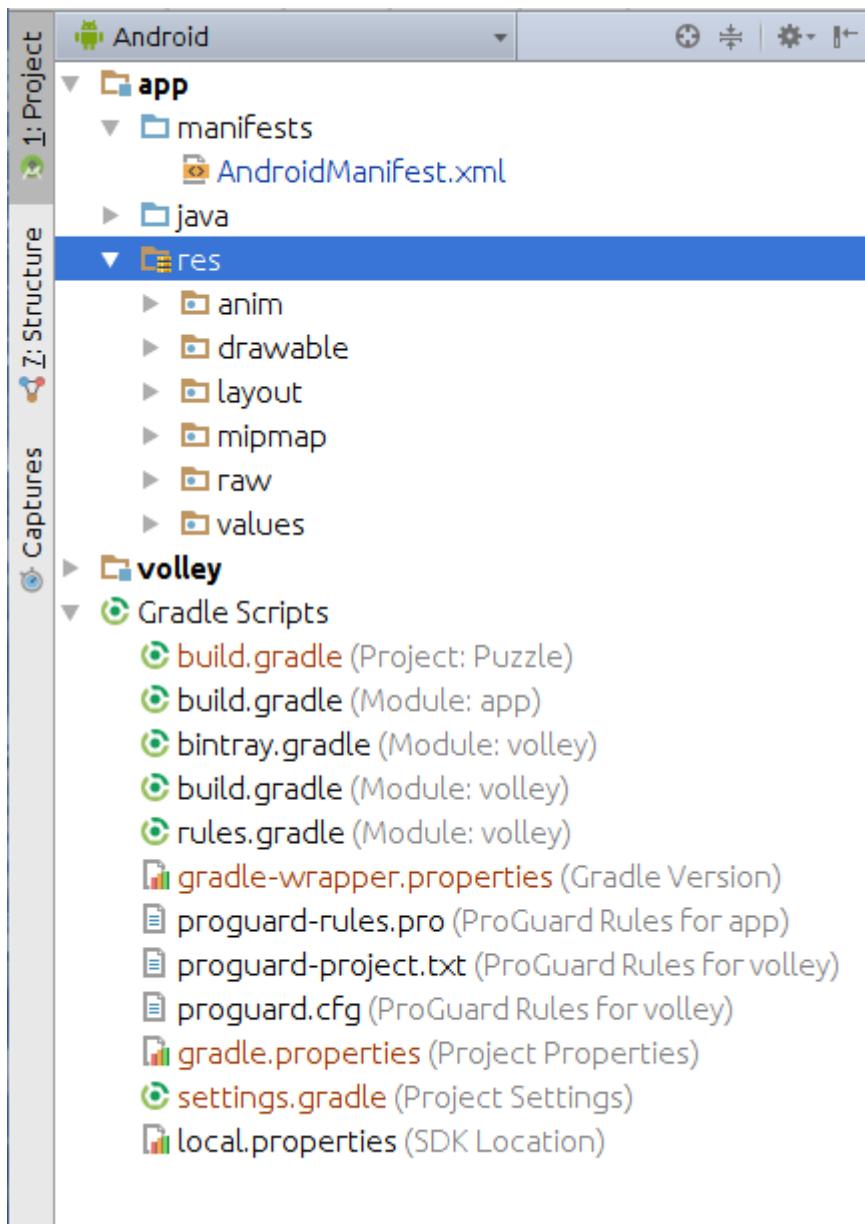


Imagen 17. Estructura proyecto Android Studio

## AndroidManifest.xml

El AndroidManifest.xml es un archivo de configuración esencial creado automáticamente por Android Studio y que revela información acerca de la aplicación y del sistema. Esta información abarca aspectos como sus capacidades y requisitos, las clases que los implementan y los datos que pueden manejar y cuándo deben ser lanzados. En este fichero también pueden concretarse permisos y políticas de seguridad que afectan a toda la aplicación.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.antonio.puzzle" >

    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />

    <uses-feature
        android:name="android.hardware.touchscreen.multitouch"
        android:required="false"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/logo2"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.DayNight.NoActionBar">
        <activity android:name=".Ini_screen"
            android:screenOrientation="landscape">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".MainActivity"
            android:screenOrientation="landscape"></activity>

        <activity android:name=".Level"
            android:screenOrientation="landscape"></activity>

        <activity android:name=".Elegir_nivel"
            android:screenOrientation="landscape"></activity>

        <activity android:name=".RegisterActivity"
            android:screenOrientation="landscape"></activity>

        <activity android:name=".LoginActivity"
            android:screenOrientation="landscape"></activity>

        <activity android:name=".pruebas_vel"
            android:screenOrientation="landscape"></activity>

        <activity android:name=".Registro_datos"
            android:screenOrientation="landscape"></activity>

        <activity android:name=".Fondo"
            android:screenOrientation="landscape"></activity>

        <activity android:name=".Preferencias"
            android:screenOrientation="landscape"></activity>
    
```

```

<activity android:name=".AudioRecordTest"
    android:screenOrientation="landscape"></activity>

<activity android:name=".Juegos"
    android:screenOrientation="landscape"></activity>

<activity android:name=".Juegos_2"
    android:screenOrientation="landscape"></activity>

</application>

</manifest>

```

El archivo AndroidManifest este programado en lenguaje xml que se constituye por etiquetas. La etiqueta de mayor tamaño es <manifest> y tiene como atributos xmlns:android que especifica el espacio de nombres para el documento. Esta etiqueta permite también hace referencia a términos relacionados con Android por todo el documento XML. El elemento manifest establece tres propiedades:

- El atributo package se utiliza para especificar el nombre del paquete primario.
- El atributo android: versionCode indica la última versión actualizada por el programador. Este valor se utiliza por parte del sistema a efectos de actualizaciones.
- El atributo android: versionName puede seguir cualquier regla que deseemos. Este valor se muestra al usuario pero no tiene ningún efecto en el sistema.

La etiqueta <application> permite configurar aspectos básicos de la aplicación, como son el icono, el tema, el label o la orientación por defecto de nuestra aplicación (horizontal en este caso). La etiqueta <application> contiene todas las actividades que forman nuestra aplicación.

El atributo android: name incluye el lugar donde se encuentra la Activity. La actividad con valor LAUNCHER en su etiqueta <category> denota que actividad será la primera en lanzarse al pulsar el icono de la aplicación

Un permiso es una restricción que limita el acceso a parte del código o a parte de la información del dispositivo. Cada permiso se identifica por una etiqueta única. Alguna de los permisos más utilizado son:

```

<uses-permission
    android:name="android.permission.INTERNET" />
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
    android:name="android.permission.RECORD_AUDIO" />

```

Si nuestra aplicación necesita permiso para utilizar algunas de las características no permitidas, debe declarar su permiso en el Manifest mediante la etiqueta <uses-permission>. También pueda proteger determinada información obtenida a través de la aplicación con la etiqueta <permission> [18].

## Gradle

Gradle es el compilador o constructor de Android. Gradle se encarga de compilar todos los archivos (.java o .xml) en un archivo comprimido APK (extensión para archivos Android).

Android studio utiliza Gradle como herramienta para automatizar y administrar el proceso de compilación. También permite definir configuraciones de compilaciones personalizadas, flexibles y reutilizar partes comunes de código en todas las versiones de la app.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 24
    buildToolsVersion "24.0.0"
    useLibrary 'org.apache.http.legacy'

    defaultConfig {
        applicationId "com.example.antonio.puzzle"
        minSdkVersion 15
        targetSdkVersion 24
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-
android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:24.0.0'
    compile project(':volley')
    compile project(path: ':volley')
}
```

En el proceso de compilación los compiladores convierten el código fuente en archivos DEX, los cuales incluyen el código de bytes que se ejecutan en dispositivos Android. El empaquetador de APK combina los archivos DEX y los recursos compilados en un solo APK. A continuación, el empaquetador APK firma el APK usando el keystore. Por último el empaquetador usa la herramienta zipalign para optimizar tu app de modo que use menos memoria cuando se ejecute en un dispositivo [21].

## Activities

Una Activity es una función que interactúa con el usuario. Dentro de una Activity el usuario puede definir la interfaz de usuario con el método **setContentView**. Normalmente una aplicación Android está compuesta por varias Activities. La primera Activity que aparece al iniciar la aplicación se conoce como **MainActivity** y a partir de ella se pueden crear el resto de Activities [19].

Cada vez que iniciamos una Activity, la Activity anterior pasa a un estado de parado pero se almacena en una pila con mecanismo “last in, first out”, de modo que cuando el usuario pulsa el botón de retroceso, la actividad anterior se relanza.

Para iniciar una Activity solo tenemos que crear una instancia que describa la actividad que queremos crear y a continuación llamar a la función **StartActivity**.

```
newActivity.setContentView(R.layout.activity_main);  
Intent intent = new Intent(getApplicationContext(), MainActivity.class);  
newActivity.startActivity(intent);
```

Las Activities cuentan con una serie de estados principales que determinan lo que se conoce como su ciclo de vida. Los estados principales de una Activity son:

**Running:** Si la actividad se encuentra en la cima de la pila, entonces está activa. La actividad se encuentra en estado activo y visible para el usuario.

**Paused:** La actividad está activa pero no es visible para el usuario. La actividad mantiene todos los cambios que el usuario haya realizado anteriormente.

**Stopped:** La actividad se encuentra parada, sin embargo la información del usuario sigue almacenada en memoria.

Si una actividad se encuentra en estado parado o pausado, el sistema podría borrar estas actividades para hacer que pasen al estado *finish*.

El ciclo de vida de una actividad Android comienza con la primera llamada al método **onCreate**, donde se crea el estado global de la aplicación. Se crean las variables estáticas, las vistas y se refresca el estado previo de la actividad en caso de existir. Una actividad termina su ciclo de vida llamando al método **onDestroy**, liberando todos los recursos creados [20].

El ciclo de vida visible de una actividad ocurre entre el método **onStart** y el método **onStop**. Durante este intervalo podemos esconder o hacer visibles recursos de la sesión de un usuario.

El ciclo de vida de primer plano ocurre entre el método **onResume** y **onPause**. Cuando llamamos al método **onPause** la actividad pasa a un segundo plano. Es común que una actividad pase de un estado a otro con frecuencia por eso es importante que estos métodos contengan pocas líneas de código.

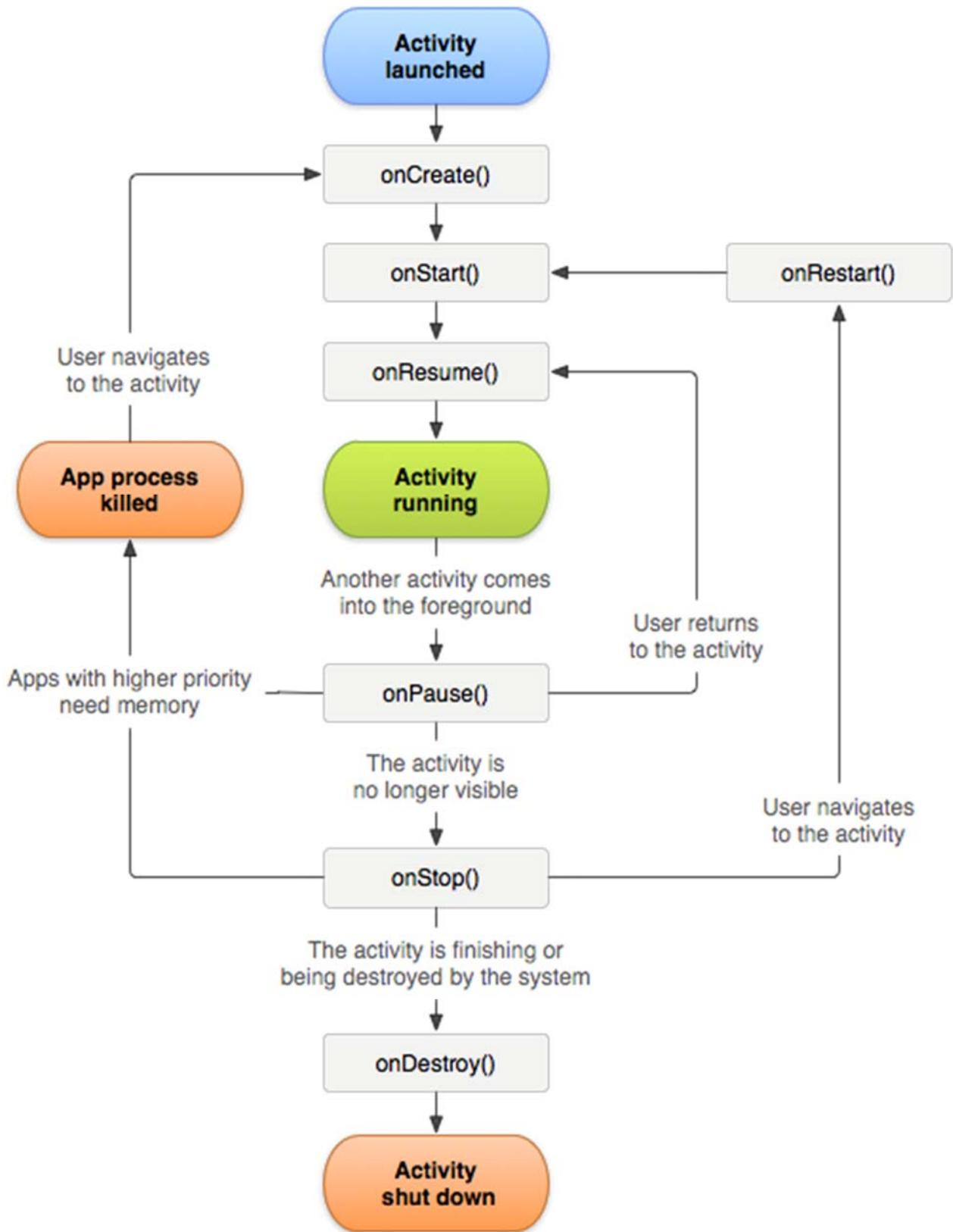


Imagen 18. Ciclo de vida de una Activity

Cada vez que una Activity pasa de un estado a otro, implementa alguno de los métodos responsables de dicho cambio. Los métodos son los siguientes:

```
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    // The activity is being created.  
}  
@Override  
protected void onStart() {  
    super.onStart();  
    // The activity is about to become visible.  
}  
@Override  
protected void onResume() {  
    super.onResume();  
    // The activity has become visible (it is now "resumed").  
}  
@Override  
protected void onPause() {  
    super.onPause();  
    // Another activity is taking focus (this activity is about to be "paused").  
}  
@Override  
protected void onStop() {  
    super.onStop();  
    // The activity is no longer visible (it is now "stopped")  
}  
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    // The activity is about to be destroyed.  
}
```

Imagen 19. Métodos ciclo de vida Activity

## Layouts

Los layouts definen la estructura visual de una interfaz de usuario para una Activity. Android Studio utiliza el lenguaje XML para implementar los layouts. El lenguaje XML permite crear interfaces gráficas de usuario de forma similar al lenguaje HTML. Ambos lenguajes utilizan etiquetas anidadas para definir cada elemento. Cada layout debe tener una etiqueta raíz View o ViewGroup y dentro de estas incluimos los elementos secundarios como Buttons, TextViews o ProgressBar [24].

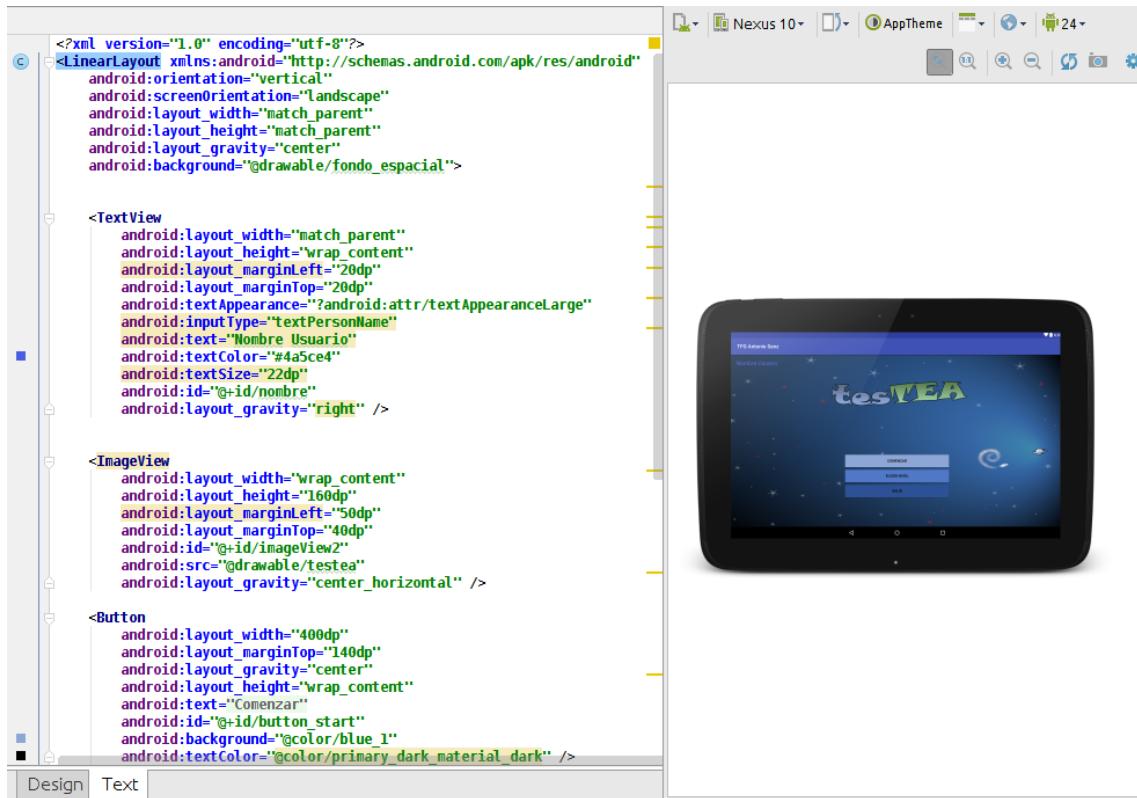


Imagen 20. Layout con su código xml

Dentro de cada elemento podemos definir atributos de posición, color, tamaño y un identificador para identificar de manera exclusiva la vista cuando hagamos una instancia de ella. El id debe ser único dentro del layout donde este implementado aunque también conviene que sea totalmente único entre todos los layouts realizados [24].

Podemos definir el ancho y alto de una vista con los atributos `layout_width` y `layout_height` pero normalmente se utilizan los atributos `wrap_content` para indicar que se modifique el tamaño conforme a los requisitos del contenido o con `match_parent` para indicar que se aumente el tamaño tanto como lo permita su grupo de vistas principal [24].

Para cargar un Layout desde la aplicación tenemos que llamar a la función `setContentView` dentro del método `onCreate` pasándole la referencia del layout.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

```

## Canvas

Canvas es una clase Java que contiene métodos principales para dibujar componentes básicos. Para realizar un dibujo son necesarios 4 componentes. Una variable **Bitmap** para registrar los pixeles, una variable Canvas como base para realizar los dibujos, el tipo de dibujo (Rect, Text, Circle, Path) y una variable Paint para indicar el color y las características del dibujo [21].

```
@Override
    public void onDraw(final Canvas canv) {

        color = registro.getFondo();

        switch(color){

            case 1: setBackgroundColor(0xffffabfd0); break; //rosa
            case 2: setBackgroundColor(0xfffbce29a); break; //verde
            case 3: setBackgroundColor(0xffbfd3fa); break; //azul
            case 4: setBackgroundColor(0xffffffff); break; //blanco
            case 5: setBackgroundResource(R.drawable.madera_1); break;
        }

        canv.drawRect(200, 120, 350, 160, whitePaint);
        canv.drawRect(200, 120, 350, 160, red_stroke);

        canv.drawRect(150, 190, 400, 230, whitePaint);
        canv.drawRect(150, 190, 400, 230, red_stroke);

        canv.drawRect(100, 260, 450, 300, whitePaint);
        canv.drawRect(100, 260, 450, 300, red_stroke);

        canv.drawRect(800, 120, 950, 160, whitePaint);
        canv.drawRect(800, 120, 950, 160, black_Stroke);

        canv.drawRect(750, 190, 1000, 230, whitePaint);
        canv.drawRect(750, 190, 1000, 230, black_Stroke);

        canv.drawRect(700, 260, 1050, 300, whitePaint);
        canv.drawRect(700, 260, 1050, 300, black_Stroke);
```

```

greenPaint = new Paint();
greenPaint.setColor(Color.GREEN);
greenPaint.setStyle(Paint.Style.FILL);

bluePaint = new Paint();
bluePaint.setColor(Color.BLUE);
bluePaint.setStyle(Paint.Style.FILL);

redPaint = new Paint();
redPaint.setColor(Color.MAGENTA);
redPaint.setStyle(Paint.Style.FILL);

yellowPaint = new Paint();
yellowPaint.setColor(Color.YELLOW);
yellowPaint.setStyle(Paint.Style.FILL);

whitePaint = new Paint();
whitePaint.setColor(Color.WHITE);
whitePaint.setStyle(Paint.Style.FILL);

blackPaint = new Paint();
blackPaint.setColor(Color.BLACK);
blackPaint.setStyle(Paint.Style.FILL);

red_stroke = new Paint();
red_stroke.setStyle(Paint.Style.STROKE);
red_stroke.setStrokeWidth(7);
red_stroke.setColor(Color.MAGENTA);

green_Stroke = new Paint();
green_Stroke.setStyle(Paint.Style.STROKE);
green_Stroke.setStrokeWidth(7);
green_Stroke.setColor(Color.GREEN);

yellow_Stroke = new Paint();
yellow_Stroke.setStyle(Paint.Style.STROKE);
yellow_Stroke.setStrokeWidth(7);
yellow_Stroke.setColor(Color.YELLOW);

blue_Stroke = new Paint();
blue_Stroke.setStyle(Paint.Style.STROKE);
blue_Stroke.setStrokeWidth(7);
blue_Stroke.setColor(Color.BLUE);

white_Stroke = new Paint();
white_Stroke.setStyle(Paint.Style.STROKE);
white_Stroke.setStrokeWidth(7);
white_Stroke.setColor(Color.WHITE);

black_Stroke = new Paint();
black_Stroke.setStyle(Paint.Style.STROKE);
black_Stroke.setStrokeWidth(7);
black_Stroke.setColor(Color.BLACK);

```

Hemos definido todas las clases Paint necesarias para dibujar las piezas de todos los puzzles, variando el color (GREEN, YELLOW, RED...), el estilo (STROKE o FILL) y el grosor del trazo.

La librería Canvas cuenta con varios métodos para dibujar figuras geométricas y/o imágenes, entre los que más hemos utilizado se encuentran:

- Círculos

```
drawCircle(float cx, float cy, float radius, Paint paint)
```

- Rectángulos y cuadrados

```
drawRect(float left, float top, float right, float bottom, Paint paint)
```

- Imágenes

```
drawBitmap(Bitmap bitmap, Matrix matrix, Paint paint)
```

## Splash Screen

La primera Activity que inicia nuestra aplicación es `Ini_screen`. La mayoría de las aplicaciones suelen tener una Activity de inicio donde se muestra el logo y el nombre de la empresa o diseñadores de la aplicación. La lógica de la clase java para esta actividad es muy sencilla. Al iniciarse mostramos el contenido del layout correspondiente y durante dos segundos se carga una barra de progreso que al finalizar lanza la siguiente actividad, en nuestro caso la pantalla de login.



Autor: Antonio Sanz Corbalán

Tutor: Antonio García Herraiz

Título: Desarrollo de una aplicación infantil para detección de anomalías en el desarrollo

*Imagen 21. Pantalla Splash*

## onTouchEvent

Hemos implementado este método en la aplicación para capturar movimientos y pulsaciones sobre la pantalla. A continuación se muestra la estructura del método `onTouchEvent` con los diferentes gestos recogidos por Android [22].

```

@Override
public boolean onTouchEvent(MotionEvent event) {

    int actionIndex = event.getActionIndex();

    switch (event.getActionMasked()) {
        //getActionMasked devuelve la acción realizada sin información
        //acerca del puntero que realiza dicha acción

        case MotionEvent.ACTION_DOWN:           //Primer puntero que toca la
                                                //pantalla
            break;

        case MotionEvent.ACTION_POINTER_DOWN:
        //Resto de punteros que tocan la pantalla
            break;

        case MotionEvent.ACTION_MOVE:
        //Después de presionar la pantalla realizamos un cambio
            break;

        case MotionEvent.ACTION_UP: //Ultimo puntero deja de tocar la
                                //pantalla.
            break;

        case MotionEvent.ACTION_POINTER_UP:
        //Resto de punteros excepto el último deja de tocar la pantalla
            break;

        case MotionEvent.ACTION_CANCEL:
            break;

        default:
            break;
    }

    return true;
}

```

## MotionEvent

MotionEvent describe los distintos tipos de movimiento y analiza la información de la acción realizada para obtener el tipo de movimiento realizado por el usuario. Cuando el usuario toca por primera vez la pantalla, el sistema detecta una acción de tipo **ACTION\_DOWN** y establece unos valores X e Y según la posición de presión con el dedo. Cada puntero tiene un identificador único que es asignado al pulsar la pantalla con **ACTION\_DOWN** y deja de ser válido hasta que dejamos de pulsar con **ACTION\_UP** o cuando la acción es cancelada con **ACTION\_CANCEL**. El identificador **ACTION\_MOVE** registra múltiples movimientos sobre un objeto y los desplaza hacia la dirección donde mantenemos la presión [23].

## Multi-Touch

Cuando tocamos la pantalla con varios dedos simultáneamente el sistema registra un primer evento **ACTION\_DOWN** para el primer puntero y un segundo evento **ACTION\_POINTER\_DOWN** para el resto de punteros. Y cuando el usuario deja de pulsar la pantalla la acción detectada es

**ACTION\_POINTER\_UP** para los primeros punteros y **ACTION\_UP** para el último puntero que deja de tocar la pantalla [24].

La clase **MotionEvent** cuenta con varios métodos para determinar la posición de los punteros en la pantalla, como **getX**, **getY** o **getAxisValue**.

Para registrar el tipo de acción detectada utilizamos el método **getActionMasked** o **getAction**, este método no reporta información acerca del índice del puntero utilizado, para registrar esta información debemos utilizar el método **getActionIndex** o **getPointerId** [28].

## Velocity Tracker

Android cuenta con una clase llamada **VelocityTracker** que permite rastrear los eventos de pulsación y movimiento, y calcular las velocidades de rastreo que realiza el usuario sobre la pantalla.

Para registrar la velocidad hemos definido un parámetro de tipo **VelocityTracker** al que debemos asociarle un movimiento llamando a la función **addMovement (MotionEvent event)**. A continuación llamamos a la función **obtain** para que el parámetro **VelocityTracker** almacene la velocidad del movimiento y a la función **computeCurrentVelocity** para que calcule la velocidad según el número de puntos registrados en la pantalla. Podemos conocer la velocidad según el eje X llamando a la función **getXVelocity** y según el eje Y llamando a **getYVelocity**. Para resetear el parámetro **VelocityTracker** utilizamos la función **clear** [25].

```
VelocityTracker tracker = null;

if (tracker == null) {
    tracker = VelocityTracker.obtain();
} else {
    tracker.clear();
}

tracker.addMovement(event);

xTouch = (int) event.getX(0);
yTouch = (int) event.getY(0);
```

## Animaciones

Existen dos formas para realizar animaciones en Android. Una manera es aplicando pequeñas transformaciones a una imagen en base a un fichero XML. La otra forma sería constituyendo una serie de imágenes similares que al reproducirse en serie cree una animación. En nuestra aplicación hemos optado por la primera opción. Hemos creado distintos ficheros XML donde definimos los cambios que vamos a aplicar sobre una imagen.

La clase mostrada a continuación define una variable de tipo **Animation** y le asigna la transformación realizada en el fichero move.xml (mostrada también a continuación). Para aplicar dicha animación sobre una imagen solamente tenemos que pasarle como argumento el nombre la View definida. Por ejemplo **move(flash\_stars)** [26].

```

public void move(View view){
    Animation animation =
    AnimationUtils.loadAnimation(getApplicationContext(),
    R.anim.move);
    view.startAnimation(animation);
}

<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/linear_interpolator"
    android:fillAfter="true">

    <translate
        android:fromXDelta="0%p"
        android:toXDelta="120%p"
        android:duration="1000" />

</set>

```

## MediaPlayer

La clase **MediaPlayer** se emplea para controlar archivos de audio. Se crea una variable de tipo **MediaPlayer** y asignamos la ruta donde se encuentra el archivo de audio. Para controlar la reproducción empleamos los métodos **start**, **pause**, **stop**, **reset** [27].

```

final MediaPlayer correct = MediaPlayer.create(this, R.raw.correct);
correct.start();

```

## AlertDialog

La clase **AlertDialog** permite crear mensajes de alerta para advertir al usuario de una acción y cerciorar la acción [28]. Para ellos creamos un objeto tipo **AlertDialog** al que hemos llamado myMessage. Una vez definido podemos incluir aspectos de diseño como un título para el dialogo con **setTitle**, hasta tres botones con distintas apariencias (**PositiveButton**, **NegativeButton** o **NeutralButton**), un icono con **setIcon** y un mensaje con **setMessage**. Por último para crear el dialogo llamamos a la función **show**.

```

AlertDialog.Builder myMessage = new AlertDialog.Builder(this);

myMessage.setTitle("Cerrar Sesión")

.setPositiveButton("Si", new DialogInterface.OnClickListener(){

@Override
public void onClick(DialogInterface dialog, int which){
    setContentView(R.layout.login);
    Intent intent = new Intent(getApplicationContext(),
    LoginActivity.class);
    startActivity(intent);
}

.setNegativeButton("No", new DialogInterface.OnClickListener(){

@Override
public void onClick(DialogInterface dialog, int which){
    dialog.dismiss();
}

.setIcon(R.drawable.alert)
.create()
.show();

```

## Comunicación entre base de datos y aplicación

La aplicación desarrollada esta comunicada con una base de datos donde registramos a los usuarios que utilizan la aplicación, junto con datos que reflejan los usuarios con la aplicación.

El modelo entidad relación de la base de datos es sencillo. Disponemos de dos tablas, una para guardar el nombre de usuario y su contraseña, y otra tabla para almacenar el nombre de usuario junto con todos los datos informativos del usuario (fallos, velocidad eje X, velocidad eje Y, jerk, tiempos...).

## Base de datos

La tabla implementada para almacenar los credenciales de los usuarios cuenta con tres variables. El nombre de usuario o username y la contraseña o password de tipo varchar con un tamaño de hasta 50 bytes. Y un identificador de usuario o user\_ID de tipo entero con un tamaño de 4 bytes y definida como clave primaria de la tabla auto incrementable.

La tabla datos almacena los datos del usuario en cada puzzle realizado. Tenemos tres variables de tipo entero N1, N2 y N3 con un tamaño de 4 bytes. Dos variables para el tiempo entre pulsaciones diffTime y para el tiempo total por puzzle timeTotal definidas tipo float con un tamaño de 6 bytes. Las variables username y date para registrar nombre, fecha y hora, ambas de tipo varchar con un tamaño de 50 y 100 caracteres respectivamente. Las variables puzzle y fallos para indicar el identificador del puzzle realizado y el número de intentos necesarios para resolverlo correctamente. Y por último dos variables masVeloX y mxsVeloY de tipo float con 6 bytes que registran una media de la velocidad de arrastre de las figuras del puzzle.

```

create table datos
(
username varchar(50),
fecha varchar(100),
puzzle int(4),
maxVeloX float(6),
maxVeloY float(6),
fallos int(4),
diffTime float(6),
timeTotal float(6),
N1 int(4),
N2 int(4),
N3 int(4)
);

create table usuarios
(
user_ID int(4) AUTO_INCREMENT PRIMARY KEY
username varchar(50),
password varchar(50)
);

```

El modelo entidad relación de la base de datos para nuestra aplicación tiene dos entidades definidas. Una entidad Usuarios que contiene la información de cada usuario en nuestro caso referida al nombre y a la contraseña. Y una entidad Datos Usuario que contiene una serie de variables que nos ayudan a identificar patrones o comportamientos inusuales en un usuario.

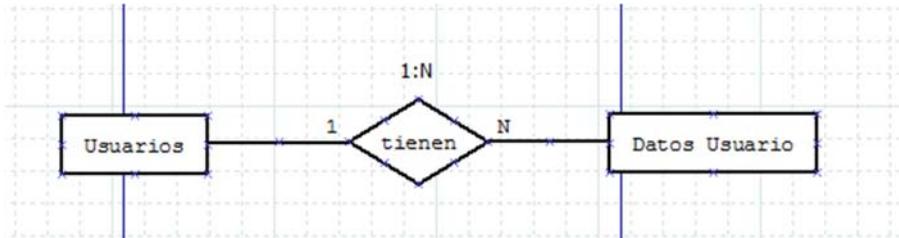


Imagen 22. Esquema BBDD modelo relacional

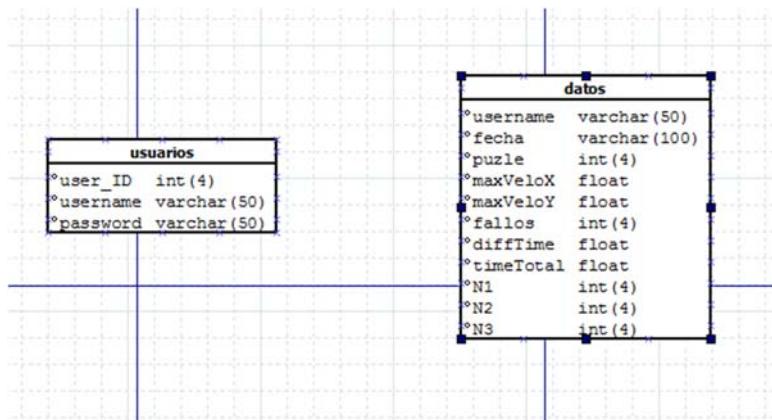


Imagen 23. Tablas de la base de datos



Imagen 24. Esquema comunicación Servidor-BBDD-App

Hemos creado una cuenta en la página de hosting gratuito hostinger para administrar la base de datos responsable de almacenar la información de cada usuario. Hostinger es un servicio de hosting gratuito que dispone de la herramienta PhpMyAdmin, con la que administramos en lenguaje MySQL las tablas y entradas de nuestra base de datos.

Para controlar el acceso a la aplicación hemos creado una tabla llamada usuarios donde almacenamos un nombre de usuario, una contraseña y un número de identificación asignado según el orden de registro por la base de datos y otra tabla donde registramos todos los datos que cada usuario registra sobre la aplicación.

## Registro de usuarios

El archivo Register es responsable de guardar los datos de registro (username y password), introducidos en la aplicación, en la base de datos.

Primero realizamos un intento de conexión con el servidor y la base de datos, en caso de error se imprime el mensaje de error. A continuación declaramos las variables username y password asignándole los valores recibidos desde la aplicación. Por último insertamos en la tabla los valores en sus respectivas columnas. Si todos los pasos se realizan correctamente, enviamos un flag ACK a la aplicación.

```

<?php
$con=mysqli_connect("mysql.hostinger.es", "u878925327_bd",
"rootadmin", "u878925327_bd"); //Establecemos la conexión

if(!$con){
    die( 'connection error: ' .mysqli.connect_error() );
}
//Definicion de variables

$username = $_POST[username];
$password = $_POST[password];

//Secuencia MySQL para introducir un usuario
$query = "INSERT INTO usuarios (username, password) VALUES (?, ?)";
$stmt = mysqli_prepare($con, $query);

if(!$stmt){
    die( 'mysqli error: ' .mysqli_error($con) );
}

mysqli_stmt_bind_param($stmt, 'ss', $username, $password);

if(mysqli_stmt_execute($stmt)){
    die( 'stmt error: ' .mysqli_stmt_error($stmt));
}

$response = array();
$response[ "success" ] = true;

echo json_encode($response);
?>

```

## Búsqueda de usuarios

El fichero FetchUserData se encarga de comprobar que los credenciales introducidos por el usuario en la aplicación (username y password) son correctos en nuestra base datos.

El proceso es inverso al anterior. En este caso definimos las variables username y password asignándole los valores introducidos por el usuario. Por otro lado realizamos un “SELECT \* FROM usuarios” que nos devuelve la tabla entera de usuario en un array. Por ultimo hacemos una comparación entre los credenciales guardados en la base de datos y los introducidos por el usuario. En caso de coincidir ambos valores, se devuelve un ACK y en caso contrario se devuelve un error.

```

<?php

$con=mysqli_connect("mysql.hostinger.es", "u878925327_bd",
"rootadmin", "u878925327_bd"); //Establecemos la conexión

$username = $_POST[username];
$password = $_POST[password];

//Secuencia MySQL para obtener un usuario con un username y un
//password
$stmt = mysqli_prepare($con, "SELECT * FROM usuarios WHERE
username = ? AND password = ?");
mysqli_stmt_bind_param($stmt, "ss", $username, $password);
mysqli_stmt_execute($stmt);

mysqli_stmt_store_result($stmt);
mysqli_stmt_bind_result($stmt, $userID, $username,
$password);

$response = array();
$response["success"] = false;

while(mysqli_stmt_fetch($stmt)){
    $response["success"] = true;
    $response["username"] = $username;
    $response["password"] = $password;
}

echo json_encode($response);
?>

```

```

02-01 20:19:29.415 23942-24411/com.example.antonio.puzzle I/System.out: open:http://www.antoniosanz.hol.es/FetchUser.php
02-01 20:19:29.417 23942-24411/com.example.antonio.puzzle I/System.out: [CDS][DNS] getAllByNameImpl netId = 0
02-01 20:19:29.417 23942-24411/com.example.antonio.puzzle D/libc-netbsd: [getaddrinfo]: hostname=www.antoniosanz.hol.es; servname=(null); cache_mode=(null),
02-01 20:19:29.417 23942-24411/com.example.antonio.puzzle D/libc-netbsd: [getaddrinfo]: ai_addrlen=0; ai_canonname=(null); ai_flags=4; ai_family=0
02-01 20:19:29.418 23942-24411/com.example.antonio.puzzle D/libc-netbsd: [getaddrinfo]: hostname=www.antoniosanz.hol.es; servname=(null); cache_mode=(null),
02-01 20:19:29.418 23942-24411/com.example.antonio.puzzle D/libc-netbsd: [getaddrinfo]: ai_addrlen=0; ai_canonname=(null); ai_flags=1024; ai_family=0
02-01 20:19:29.433 23942-23970/com.example.antonio.puzzle D/Surface: Surface::setBuffersDimensions(this=0x7f77e2c200, w=800, h=1280)
02-01 20:19:29.452 23942-23970/com.example.antonio.puzzle D/Surface: Surface::setBuffersDimensions(this=0x7f77e2c200, w=800, h=1280)
02-01 20:19:29.471 23942-23970/com.example.antonio.puzzle D/Surface: Surface::setBuffersDimensions(this=0x7f77e2c200, w=800, h=1280)
02-01 20:19:29.489 23942-23970/com.example.antonio.puzzle D/Surface: Surface::setBuffersDimensions(this=0x7f77e2c200, w=800, h=1280)
02-01 20:19:29.508 23942-23970/com.example.antonio.puzzle D/Surface: Surface::setBuffersDimensions(this=0x7f77e2c200, w=800, h=1280)
02-01 20:19:29.535 23942-24411/com.example.antonio.puzzle D/libc-netbsd: getaddrinfo: www.antoniosanz.hol.es get result from proxy >>
02-01 20:19:29.535 23942-24411/com.example.antonio.puzzle I/System.out: propertyValue:true
02-01 20:19:29.536 23942-24411/com.example.antonio.puzzle I/System.out: [CDS]rx timeout:2500
02-01 20:19:29.536 23942-24411/com.example.antonio.puzzle I/System.out: [socket][0] connection www.antoniosanz.hol.es/185.28.20.93:80;LocalPort=48054(2500)
02-01 20:19:29.536 23942-24411/com.example.antonio.puzzle I/System.out: [CDS]connect[www.antoniosanz.hol.es/185.28.20.93:80] tm:<
02-01 20:19:29.537 23942-24411/com.example.antonio.puzzle D/Posix: [Posix_connect Debug]Process com.example.antonio.puzzle :80
02-01 20:19:29.599 23942-24411/com.example.antonio.puzzle I/System.out: [socket][1/192.168.1.36:48054] connected
02-01 20:19:29.600 23942-24411/com.example.antonio.puzzle I/System.out: [OkHttp sendRequest]>
02-01 20:19:29.600 23942-24411/com.example.antonio.puzzle I/System.out: [OkHttp sendRequest]<<
```

Imagen 25. Logs del login de usuario

La imagen anterior muestra los logs en el momento de realizar un login desde la aplicación. La aplicación establece una comunicación con el fichero FetchUser de nuestro host antoniosanz.hol.es con dirección ip 185.28.29.93. A continuación envía una petición de login HTTP por el puerto 80 y espera el ACK que confirme la validez del login y lo deniegue.

## Registro de datos de usuario

El fichero DataUsuario guarda los datos registrados en cada puzzle realizado por el usuario en la aplicación. El proceso es el mismo que al realizar un registro. Primero establecemos la conexión con el servidor, declaramos todas las variables asignándoles los valores recibidos desde la aplicación, luego insertamos en la base de datos cada variable definida y por último enviamos un ACK si todo se ha realizado correctamente.

```
<?php
$con=mysqli_connect("mysql.hostinger.es", "u878925327_bd",
"rootadmin", "u878925327_bd"); //Establecemos la conexión

//Establecemos la conexión con a base de datos

if(!$con){
    die('connection error: ' .mysqli.connect_error());
}

//Definicion de variables

$maxVeloY = $_POST[maxVeloX]; //Aceleracion media ejeX
$maxVeloX = $_POST[maxVeloY]; //Aceleracion media ejeY
$fallos = $_POST[fallos]; //numero de fallos
$diffTime = $_POST[diffTime]; //tiempo entre pulsaciones
$timeTotal = $_POST[timeTotal]; //tiempo total en puzzle
$username = $_POST[username]; //nombre de usuario
$fecha = $_POST[fecha]; //fecha
$puzzle = $_POST[puzzle]; //numero de puzzle
$N1 = $_POST[N1]; //valores jerk
$N2 = $_POST[N2];
$N3 = $_POST[N3];

//Secuencia MySQL para introducir datos de un usuario en la
//tabla
$query = "INSERT INTO datos (username, fecha, puzzle, maxVeloX,
maxVeloY, fallos, diffTime, timeTotal, N1, N2, N3) VALUES (?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?)";

$stmt = mysqli_prepare($con, $query);

if(!$stmt){
    die('mysql error: ' .mysqli_error($con));
}

mysqli_stmt_bind_param($stmt, "ssiddiddiii", $username,
$fecha, $puzzle, $maxVeloX, $maxVeloY, $fallos, $diffTime,
$timeTotal, $N1, $N2, $N3);

if(mysqli_stmt_execute($stmt)){
    die('stmt error: ' .mysqli_stmt_error($stmt));
}

// Enviamos un mensaje ACK a la aplicación
$response = array();
$response["success"] = true;

echo json_encode($response);
?>
```

La aplicación debe mantener una comunicación con el servidor y la base de datos donde almacenamos toda la información sobre los usuarios como por ejemplo contraseñas, nombres de usuarios y datos de los puzzles realizados. Para realizar las conexiones Http con el servidor hemos añadido la librería Android Volley. Volley es un cliente Http que se encarga de facilitar las comunicaciones de red en aplicaciones Android. Las ventajas de Volley es que permite realizar procesamientos concurrentes de peticiones, gestión de peticiones en segundo plano e información detallada del estado de las peticiones.

El modo de implementar la comunicación entre la aplicación y el servidor es el siguiente:

1. Abrimos una conexión HTTP.
2. Creamos un objeto HttpPOST con la dirección url del servidor.
3. Creamos un objeto JSON con los datos.
4. Convertimos el objeto JSON a string.
5. Enviamos el objeto JSON al método RequestQueue de la clase Volley, que se encarga de gestionar la cola de peticiones.

```
public class LoginRequest extends StringRequest {  
    private static final String LOGIN_REQUEST_URL =  
        "http://www.antoniosanz.hol.es/FetchUser.php";  
  
    private Map<String, String> params;  
  
    public LoginRequest(String username, String password,  
        Response.Listener<String> listener) {  
        super(Method.POST, LOGIN_REQUEST_URL, listener, null);  
  
        params = new HashMap<>();  
        params.put("username", username);  
        params.put("password", password);  
    }  
  
    @Override  
    public Map<String, String> getParams() {  
        return params;  
    }  
}
```

```

public class RegisterRequest extends StringRequest {
    private static final String REGISTER_REQUEST_URL =
    "http://www.antoniosanz.hol.es/Registro.php";

    private Map<String, String> params;

    public RegisterRequest(String username, String password,
    Response.Listener<String> listener) {
        super(Method.POST, REGISTER_REQUEST_URL, listener, null);
        params = new HashMap<>();
        params.put("username", username);
        params.put("password", password);
    }

    @Override
    public Map<String, String> getParams() {
        return params;
    }
}

```

## Diseño página web

En el servidor web gratuito de hostinger he colgado una tabla donde muestro los datos registrados en la base de datos y así poder mostrarlo en un formato más claro y limpio. La URL de la página es [antoniosanz.hol.es](http://www.antoniosanz.hol.es) .

Para trasladar los datos desde la base de datos a la tabla de la página web utilizamos el siguiente modelo de diseño software para gestión de archivos y bases de datos (Data Access Object):

```

<?php

class ModeloDAO{

    private $mysqli;

    function __construct(){
        require( "Database.php" );
        $db = Database::getInstance();
        $this->mysqli = $db->getConnection();
    }

    //Mostrar usuarios
    function mostrarUsuarios(){

        $sentencia = "SELECT username, puzzle, fecha, maxVeloX, maxVeloY,
        fallos, diffTime, timeTotal, N1, N2, N3 FROM datos";

        $result = $this->mysqli->query($sentencia) or trigger_error($this-
        >mysqli->error);

        $i=0;
        $resultado = array();
        while($row = $result->fetch_array()) {
            $resultado[] = $row;
            $i++;
        }
        return $resultado;
    }

}

?>

```

Primero nos conectamos a la base de datos utilizando el modelo singleton. El modelo Singleton es un patrón de diseño de la programación orientado a objetos que consiste en tener definido una única instancia global de cada clase y crear nuevas instancias a partir de este con el método **getInstance**. A continuación definimos la función **mostrarUsuarios**, donde creamos la query para obtener aquellos datos que vayamos a mostrar en la tabla y los guardamos en un array llamado **\$resultado**.



Imagen 26. UML diagrama Singleton

```

<?php
/*
 * Mysql database class - only one connection allowed
 */
class Database {
    private $_connection;
    private static $_instance; //The single instance
    private $_host = "mysql.hostinger.es";
    private $_username = "u878925327_bd";
    private $_password = "rootadmin";
    private $_database = "u878925327_bd";

    /*
     * Get an instance of the Database
     @return Instance
    */
    public static function getInstance() {
        if(!self::$_instance) { // If no instance then make one
            self::$_instance = new self();
        }
        return self::$_instance;
    }

    // Constructor
    private function __construct() {
        $this->_connection = new mysqli($this->_host, $this->_username,
                                         $this->_password, $this->_database);
        // Error handling
        if(mysqli_connect_error()) {
            trigger_error("Failed to conencto to MySQL: " .
mysql_connect_error(),
                          E_USER_ERROR);
        }
    }

    // Magic method clone is empty to prevent duplication of
    connection
    private function __clone() { }

    // Get mysqli connection
    public function getConnection() {
        return $this->_connection;
    }
}
?>

```

## Página web



The screenshot shows a Microsoft Internet Explorer browser window with the title bar "antoniosanz.hol.es". The address bar also displays "antoniosanz.hol.es". The toolbar includes standard buttons for back, forward, search, and refresh. Below the toolbar, the status bar shows "Antonio" and the date/time "25/02/2017 11:08". The main content area features a large title "Registro de Usuarios" in bold black font. Below it is a table with 11 columns and 9 rows of data. The columns are labeled: Usuario, N° Puzzle, Fecha, maxVeloX (px/s), maxVeloY (px/s), fallos, Tiempo entre Pulsaciones (ms), Tiempo Total (seg), N1, N2, and N3. The data rows represent different puzzle attempts by various users.

| Usuario  | Nº Puzzle | Fecha               | maxVeloX (px/s) | maxVeloY (px/s) | fallo | Tiempo entre Pulsaciones (ms) | Tiempo Total (seg) | N1  | N2 | N3 |
|----------|-----------|---------------------|-----------------|-----------------|-------|-------------------------------|--------------------|-----|----|----|
| Invitado | 1         | 2017-02-15 19:39:57 | 0.142028        | 317.874         | 0     | 1095.86                       | 6.797              | 24  | 44 | 4  |
| antonio  | 1         | 2017-02-15 19:43:35 | 72.6383         | 443.205         | 0     | 1191.22                       | 9.625              | 57  | 33 | 22 |
| antonio  | 1         | 2017-02-15 19:44:56 | 1064.86         | 654.822         | 0     | 2047.2                        | 19.801             | 231 | 59 | 18 |
| Invitado | 1         | 2017-02-18 18:24:24 | 276.919         | 185.235         | 0     | 1331.57                       | 8.476              | 70  | 28 | 23 |
| Invitado | 1         | 2017-02-19 14:19:25 | 95.563          | 401.504         | 0     | 1001                          | 7.414              | 83  | 21 | 33 |
| Invitado | 1         | 2017-02-25 10:39:41 | 0.271051        | 279.335         | 0     | 1200.86                       | 7.799              | 37  | 24 | 41 |
| Invitado | 2         | 2017-02-25 10:44:24 | 138.611         | 491.452         | 0     | 1705.67                       | 0                  | 107 | 26 | 29 |
| Invitado | 2         | 2017-02-25 10:45:24 | 118.034         | 656.637         | 0     | 1352.2                        | 0                  | 102 | 7  | 26 |
| Invitado | 1         | 2017-02-25 10:50:15 | 27.8459         | 233.868         | 0     | 1098.57                       | 7.14               | 44  | 32 | 28 |

Imagen 27. Captura antoniosanz.hol.es

Hemos creado una página web para mostrar los datos registrados en la base de datos con un formato más sencillo y claro de leer. Los datos se muestran en una tabla de 11 columnas, una para cada variable registrada y una fila por cada puzzle realizado. La url de la página es [antoniosanz.hol.es](http://antoniosanz.hol.es). La estructura de la tabla está hecha con HTML y el diseño con CSS. Para acceder a la información de la base de datos desde la página web utilizamos un patrón de diseño DAO (Data Access object). De esta forma encapsulamos el acceso a los datos sin preocuparnos de cómo realizar el acceso en cada momento.

## Código HTML y PHP:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="estilos.css">
<script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
<script type="text/javascript"
src="js/jquery.tablesorter.min.js"></script>
</head>
<body>

<h1>Registro de Usuarios</h1>

<div id="wrapper">

    <table id="keywords">
        <thead>
            <tr>
                <th>Usuario</th>
                <th>Nº Puzzle</th>
                <th>Fecha</th>
                <th>maxVeloX (px/s)</th>
                <th>maxVeloY (px/s)</th>
                <th>fallos</th>
                <th>Tiempo entre Pulsaciones (ms)</th>
                <th>Tiempo Total (seg)</th>
                <th>N1</th>
                <th>N2</th>
                <th>N3</th>
            </tr>
        </thead>
        <tbody>

<?php
require("ModeloDAO.php");
$modelo = new ModeloDAO();
$datos = $modelo->mostrarUsuarios();

for($i=0; $i<count($datos); $i++)
{
    // $cad = "<tr><td>". $datos[$i][0]. "</td>" ;
    $cad = "<tr><td>". $datos[$i][0]. "</td>" ;
    $cad .= "<td>". $datos[$i][1]. "</td>" ;
    $cad .= "<td>". $datos[$i][2]. "</td>" ;
    $cad .= "<td>". $datos[$i][3]. "</td>" ;
    $cad .= "<td>". $datos[$i][4]. "</td>" ;
    $cad .= "<td>". $datos[$i][5]. "</td>" ;
    $cad .= "<td>". $datos[$i][6]. "</td>" ;
    $cad .= "<td>". $datos[$i][7]. "</td>" ;
    $cad .= "<td>". $datos[$i][8]. "</td>" ;
    $cad .= "<td>". $datos[$i][9]. "</td>" ;
    $cad .= "<td>". $datos[$i][10]. "</td></tr>" ;

    echo "$cad";
}
?>
```

```

        </tbody>
    </table>
</div>
<footer>
<p>Antonio Sanz Corbalán<br>
Detección de anomalías de espectro autista en niños a temprana
edad</p>

</footer>

<script type="text/javascript">
$(function(){
    $('#keywords').tablesorter();
});
</script>
</script>
</body>
</html>

```

Código CSS:

```

body{
    background: linear-gradient(to left, #485563 , #29323c);
}

h1{
    font-family: miFuente2;
    padding-left: 20px;
    padding-bottom: 30px;
    font-size: 54px;
    text-align: center;
    text-shadow: 4px 4px black;
    color: white;
}

p{
    font-family: miFuente2;
    color: white;
    text-align:center;
    font-size: 14px;
    padding-top: 40px;
}

```

```

@font-face{
    font-family: miFuente2;
    src: url(fuente2.otf);
}

table{
    padding-left: 5px;
    text-align: center;
    font-family: miFuente2;
    font-size: 18px;
    border-collapse: collapse;
}

tr{
    height: 50px;
    width: 200px;
}

th{
    background-color: #c99e10;
    border: black solid 1px;
    box-shadow: 1px 1px black;
}
th:hover{

}

tr:nth-child(even){
    background-color: #9b4f0f;
}

tr:nth-child(odd){
    background-color: #9b4f0f;
}

td:nth-child(2){

    width: 220px;
}

td:nth-child(3){

    width: 220px;
}

td{
    width: 140px;
    border: 1px solid black;
    color: white;
}

```

# Manual de usuario

## Instalación Android Studio

Para instalar Android Studio previamente debemos instalar Java Development Kit (JDK) para que funcione correctamente. JDK incluye muchas herramientas de Android Studio como el compilador javac o el generador de documentos javadoc. Podemos descargar JDK en este enlace [www.oracle.com/technetwork/java/javase/downloads/index.html](http://www.oracle.com/technetwork/java/javase/downloads/index.html). Asegurarse de descargar la versión correspondiente con nuestro ordenador. Una vez descargado el fichero de instalación, lo ejecutamos.

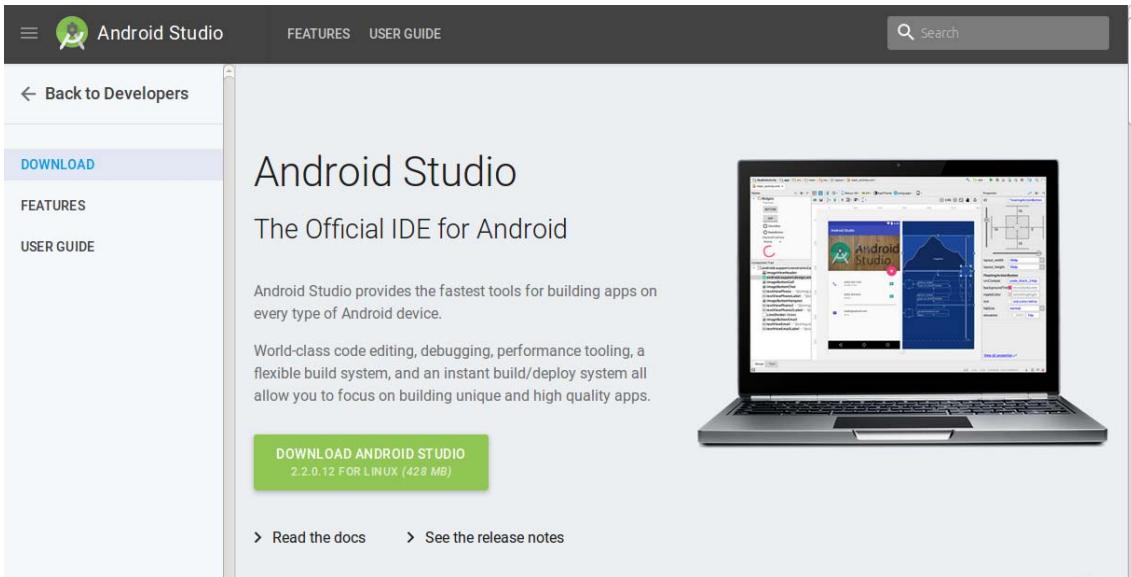


Imagen 28. Instalación Android Studio

Una vez instalado JDK podemos continuar con la instalación de Android Studio. El fichero de instalación de Android Studio lo podemos descargar en el siguiente enlace <http://developer.android.com/sdk/index.html>

Ejecutamos el fichero de instalación, aceptamos todos los acuerdos de licencia y finalizamos la instalación [29].

## Instalación Genymotion

La mejor forma de probar nuestra aplicación es con un dispositivo real, en caso de no disponer de un dispositivo real, el programa Genymotion es capaz de proporcionarnos distintos dispositivos con los que simular nuestra aplicación. Además Genymotion tiene mayor eficiencia y rendimiento y requiere de menos memoria RAM para su ejecución.

En la página <https://www.genymotion.com/#!/download> podemos descargar el fichero de instalación de Genymotion. Genymotion y Oracle Virtualbox Machine funcionan conjuntamente, de modo que se requiere la instalación de ambos programas en el ordenador para que todo funcione correctamente.

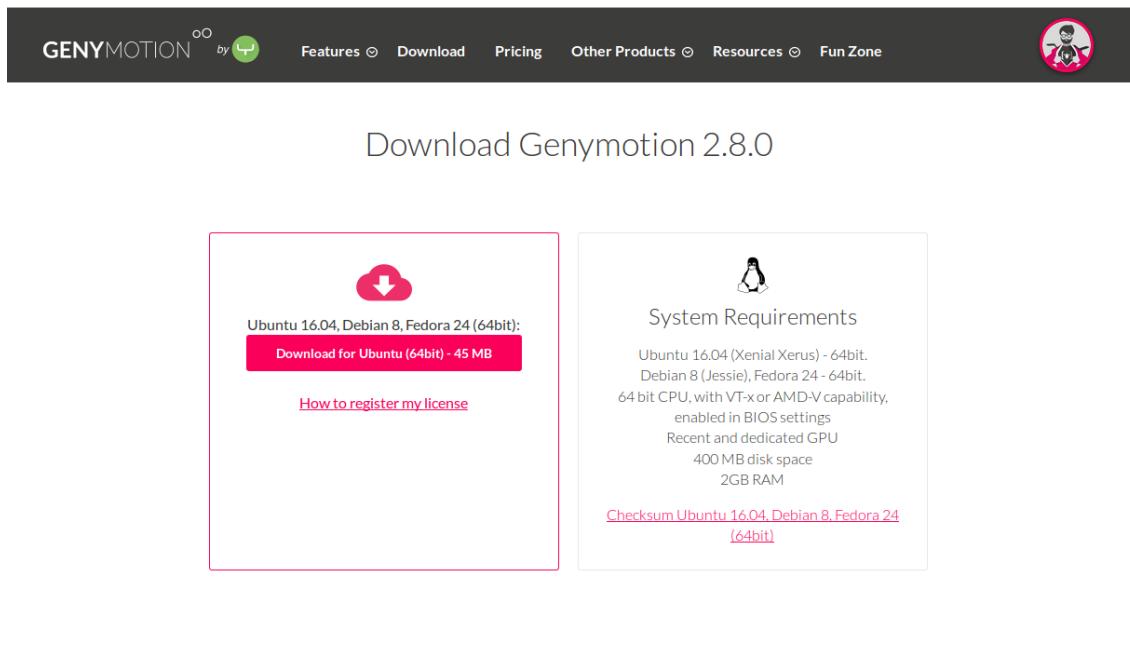


Imagen 29. Instalación Genymotion

Una vez instalado en el equipo correctamente solo tenemos que instalar el correspondiente plugin de Genymotion en Android Studio para que se active el emulador al compilar la aplicación desde Android Studio. También tenemos que elegir el tipo de dispositivo y versión Android que queremos utilizar para nuestro emulador.

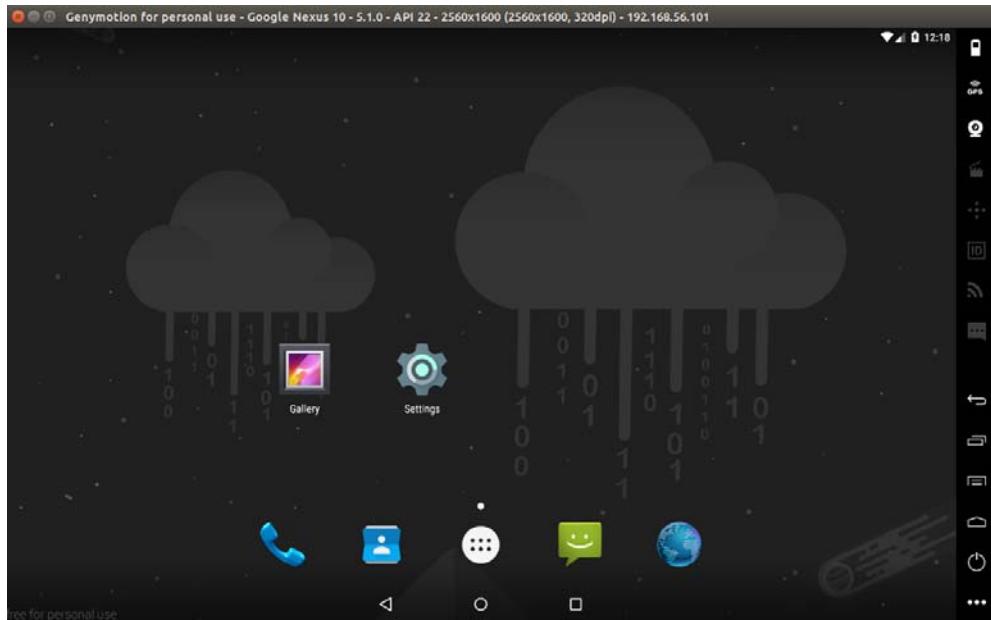


Imagen 30. Simulación con Genymotion Nexus 9

## Kobiton

Kobiton es una plataforma para testeo de aplicaciones móviles. Registrándonos en la página tenemos acceso a una gran variedad de dispositivos donde podemos instalar el apk de nuestra aplicación y realizar pruebas de testeo manuales y automáticas.

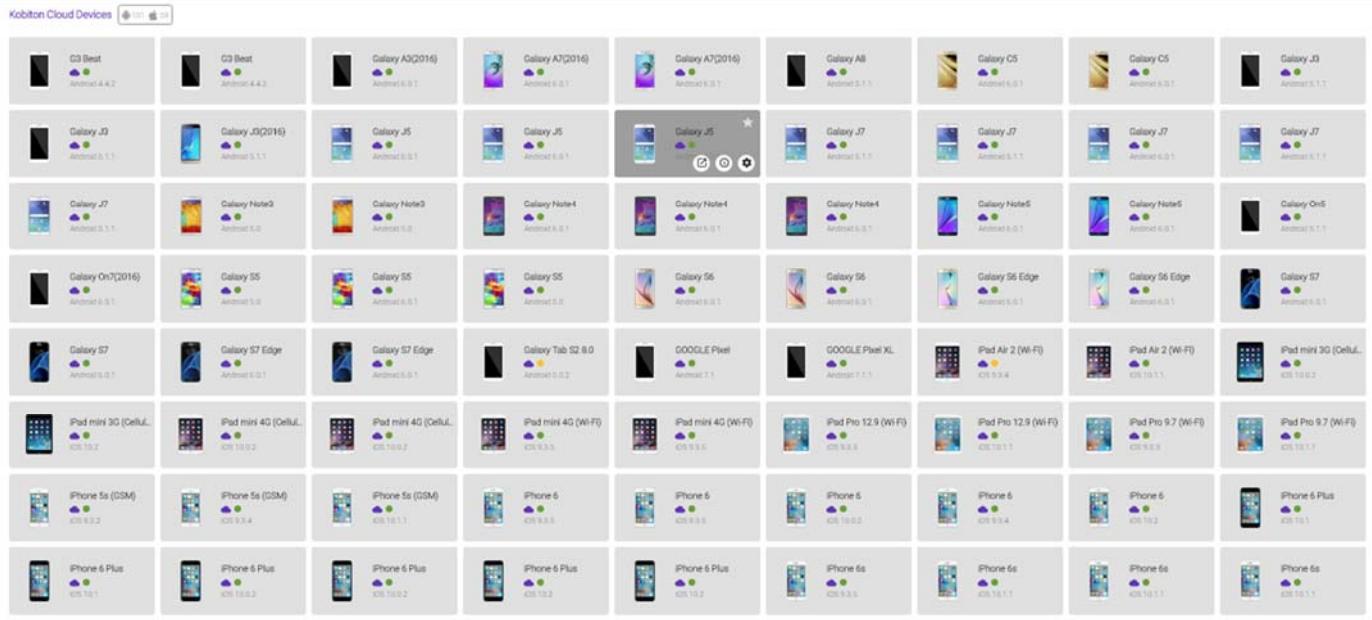


Imagen 31. Dispositivos Kobiton

Al seleccionar un dispositivo podemos controlar funcionalidades hardware del móvil como el GPS, sensores, NFC, cámara, micrófono y altavoces. Incluso podemos acceder a los logs del dispositivo para capturar procesos y rendimiento de nuestra aplicación.

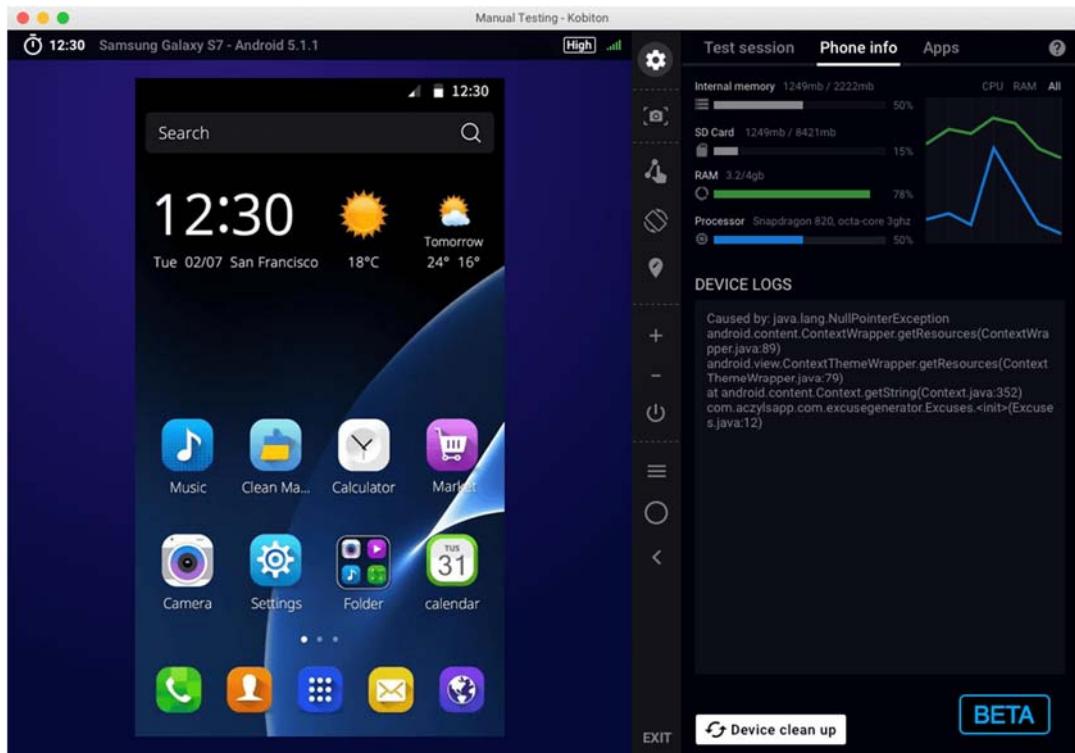


Imagen 32. Aspecto controlador dispositivo Kobiton

## Resumen funcional de la aplicación

### 1. Splash Screen



Imagen 33. Splash inicial

Al iniciar la aplicación se muestra una pantalla de progreso automático con el logo de la universidad y el título de trabajo de fin grado. Finalizado el proceso de inicio (después de 2 segundos) se muestra la pantalla de Login que da acceso a la aplicación.

## 2. Login



Imagen 34. Pantalla de logeo

A los usuarios registrados en la aplicación se les permite el acceso a la aplicación después de pulsar [Iniciar Sesión]. Para la validación de las credenciales de usuario se requiere tener conexión a red. Enviamos un objeto JSON con las cadenas de usuario y contraseña a la base datos y esta nos responda con una variable booleana indicando acierto o fallo.

### Criterios de aceptación

- Si los datos son correctos, se almacena el nombre usuario y se muestra en la parte superior izquierda de la siguiente pantalla.
- Si los datos son incorrectos, se abre una alerta de dialogo con el mensaje “Los datos introducidos no son válidos”.
- Pulsando en [Crear nuevo usuario] accedemos a la pantalla de registro de un nuevo usuario.
- Al pulsar en [Inicio como invitado] entramos en la aplicación sin una identidad de usuario.

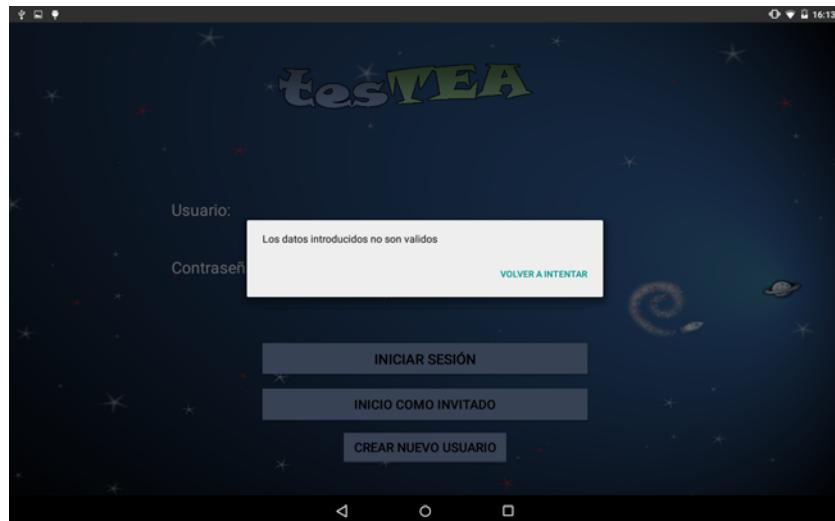


Imagen 35. Mensaje de alerta pantalla login

### 3. Sign Up



Imagen 36. Pantalla de registro

El registro de nuevos usuarios se realiza a través de esta pantalla. Introducimos un nombre de usuario y una contraseña en los campos respectivos y pulsamos el botón [Crear cuenta]. Los datos son almacenados en un fichero JSON y transmitidos a la base de datos relacionada con la aplicación.

#### Criterios de aceptación

- Si informamos el campo de nombre de usuario vacío se muestra un mensaje de error.
- Al pulsar [Volver] pasamos a la pantalla de Login.
- Al pulsar en [Crear cuenta] con usuario valido, volvemos a la pantalla de login.

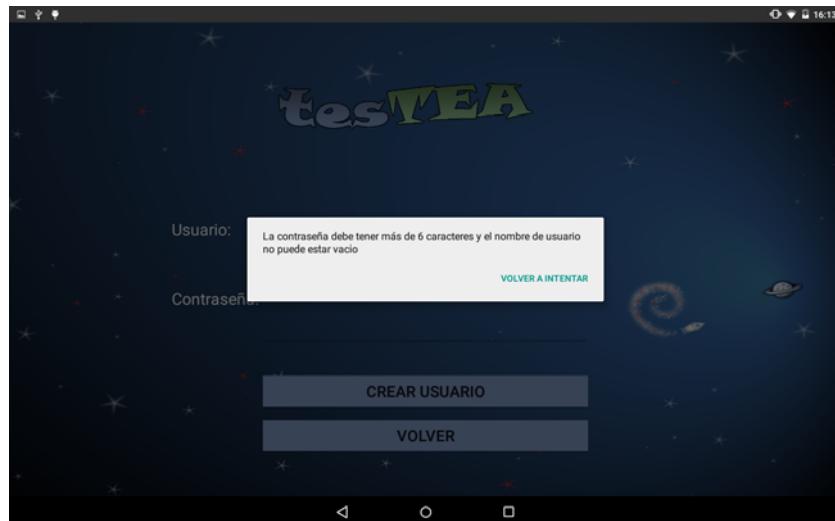


Imagen 37. Mensaje de alerta pantalla registro

#### 4. Menú

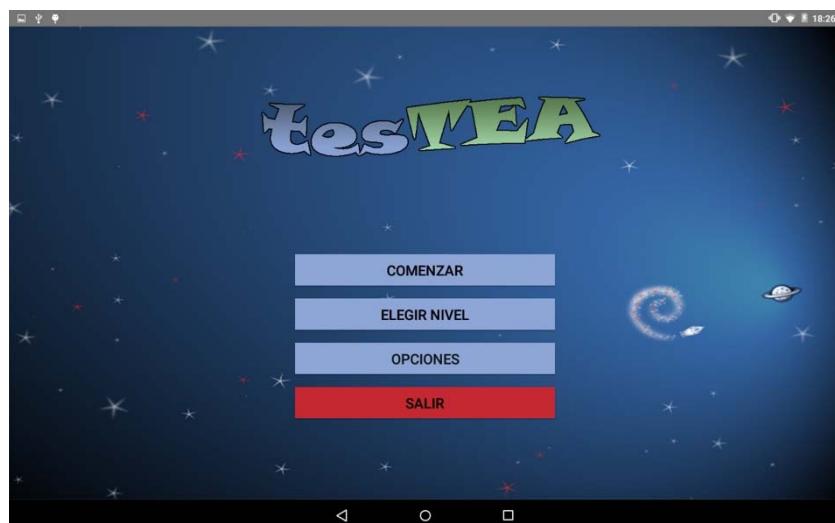


Imagen 38. Pantalla menú principal

Una vez accedido a la aplicación se muestra un menú con las distintas acciones que podemos desarrollar en la aplicación. Tenemos las opciones de [comenzar], [Elegir nivel], [opciones] y [salir].

#### Criterios de aceptación

- Al pulsar el [Comenzar] accedemos al primer puzzle registrado en la aplicación.
- Al pulsar en [Elegir nivel] accedemos a una pantalla que muestra todos los puzzles registrados en la aplicación.

- Al pulsar en [Opciones] accedemos a un submenú con las opciones de la aplicación.
- Al pulsar en [Salir] Se abre un mensaje de alerta pidiendo aserción para cerrar sesión.

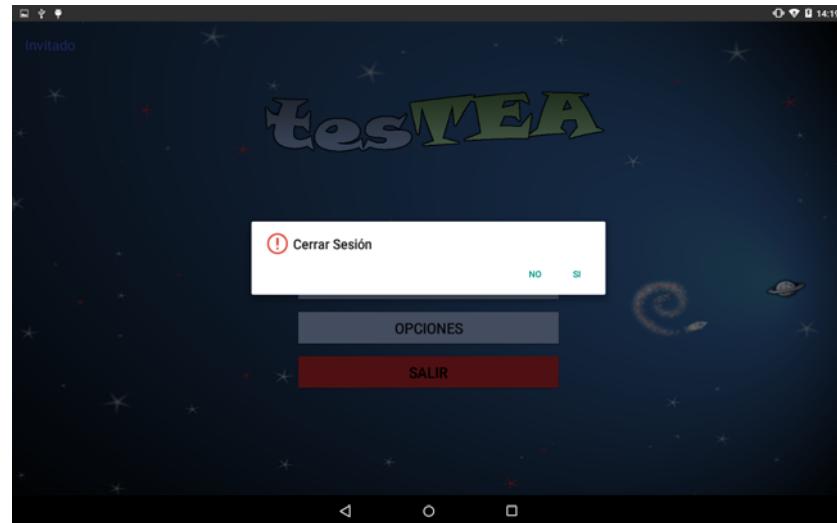


Imagen 39. Mensaje de alerta cerrar Sesión

## 5. Puzzles

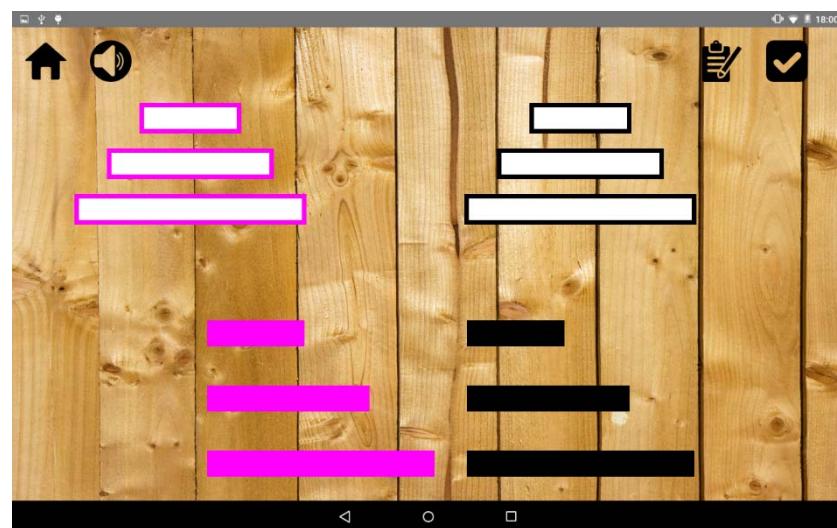


Imagen 40. Puzzle 1

Cada puzzle tiene un diseño y un rango de dificultad distinto. Al comenzar el puzzle las fichas se encuentran desordenadas en centro de la pantalla. En la parte superior izquierda se encuentran los iconos de [Volver] y [Reproducir Audio] y en la esquina superior derecha están los iconos de [Registrar] y [Comprobación].

Criterios de aceptación

- Hasta que usuario no pulsa el icono de [Registro] en la esquina superior derecha, la aplicación no comienza a registrar datos del comportamiento del usuario con la aplicación.
- Al pulsar el icono de [comprobación] situado en la esquina superior derecha, se comprueba si todas las fichas están colocadas correctamente en su posición. En caso correcto se accede a la siguiente pantalla.
- Si al pulsar el icono de [comprobación] hay alguna ficha mal colocada, se vuelven a colocar las fichas en el centro de la pantalla y el número de fallos aumenta.

## 6. Pantalla marcadora de nivel

En caso de completar un puzzle correctamente, se accede a una pantalla intermedia entre puzzles donde se realiza el envío a la base de datos de todos los datos registrados en el puzzle anterior.

### Criterios de aceptación

- La pantalla muestra el nivel superado en el centro y un grado de subnivel indicado con estrellas en la parte inferior.
- Al pulsar [siguiente nivel] accedemos al siguiente puzzle.
- Al pulsar [Salir] se muestra un mensaje de alerta preguntando si queremos volver al menú.

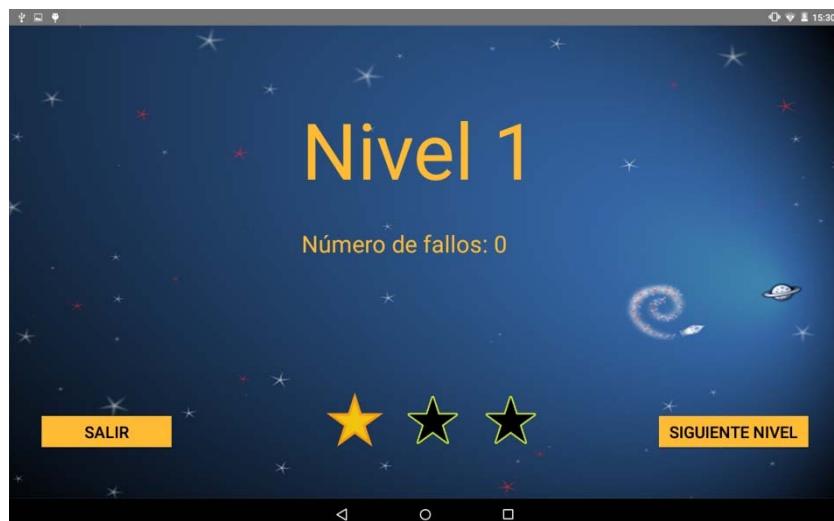


Imagen 41. Pantalla marcador nivel

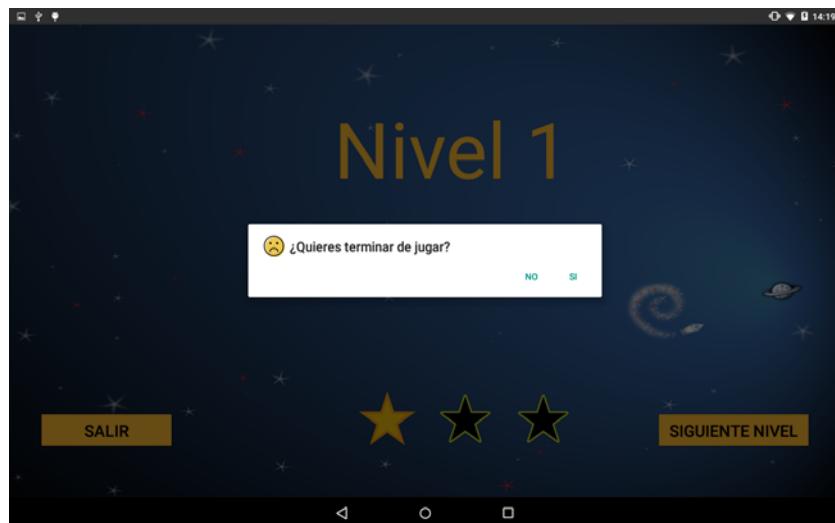


Imagen 42. Alerta salir del juego

## 7. Elegir puzzle

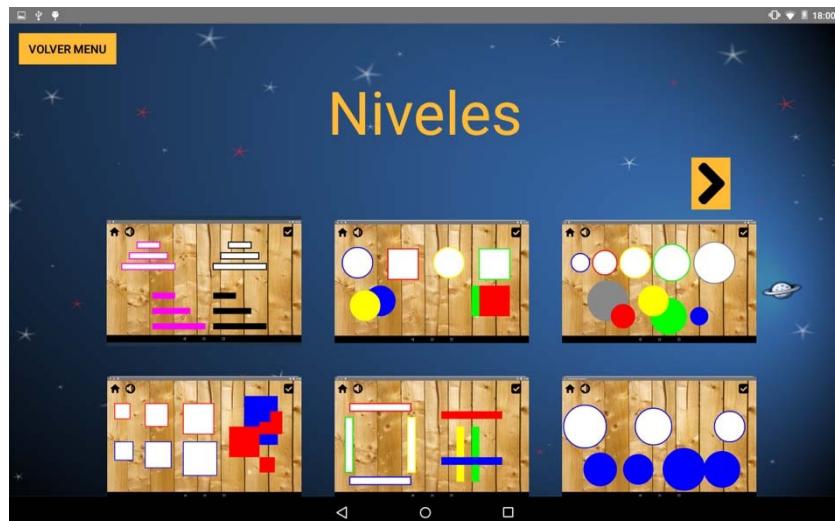


Imagen 43. Elegir puzzle

Al pulsar [Elegir puzzle] se muestra una colección de todos los puzzles guardados en la aplicación.

### Criterios de aceptación

- Al pulsar sobre la imagen de un puzzle accedemos al puzzle indicado.
- Al pulsar sobre la flecha, se muestra la siguiente página de puzzles.
- Al pulsar en [Volver al menú] accedemos al menú de la aplicación.

## 8. Opciones



Imagen 44. Menú opciones

En el submenú se encuentran las opciones de [Elegir fondo], [Grabar audio], [Medidas de Velocidad] y [Volver al Menú].

### Criterios de aceptación

- Pulsando en [Color de Fondo] accedemos a la pantalla de selección de color de fondo para el background de los puzzles.
- Pulsando en [Grabar audio] accedemos a una pantalla de grabación de audio.
- Pulsando en [Medidas de velocidad] accedemos a una pantalla para visualizar el trazado y la velocidad de las pulsaciones sobre la pantalla.
- Pulsando en [Volver a menú] accedemos al menú principal de la aplicación.

## 9. Selección de fondo

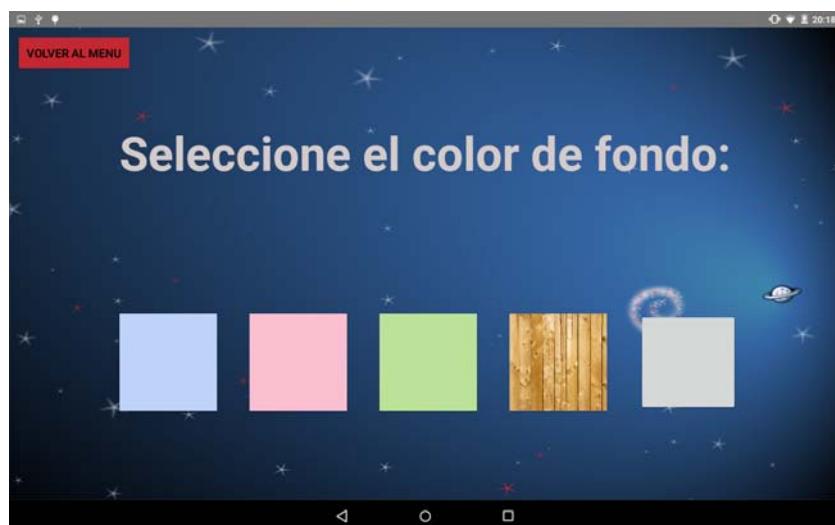


Imagen 45. Selección color fondo

Esta pantalla permite elegir entre 5 colores diferentes para el fondo. Para elegir uno de ellos hay que pulsar sobre su imagen.

#### Criterios de aceptación

- Al seleccionar uno de los fondos volvemos al menú de la aplicación y si iniciamos alguno de los puzzles, su fondo será del color seleccionado.
- Al pulsar en [Volver a menú] accedemos al menú principal de la aplicación.

## 10. Grabación de audio

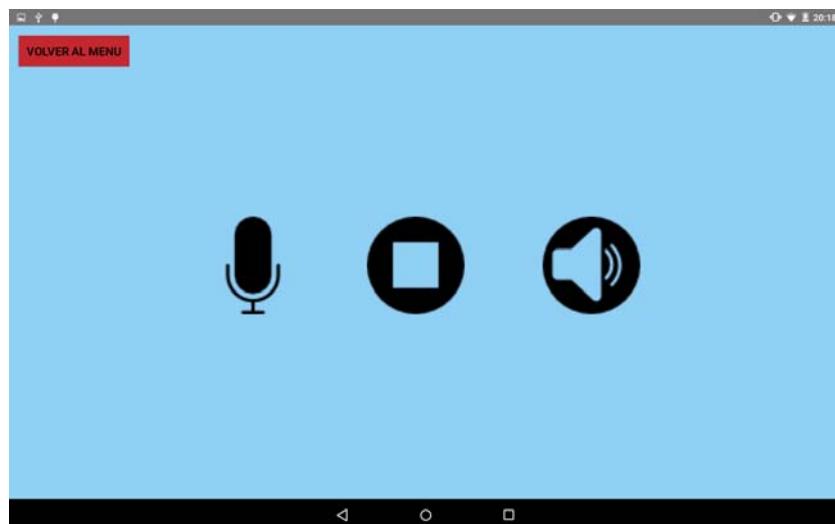


Imagen 46. Grabar Audio

La aplicación permite grabar un fichero de audio que puedan servir de instrucciones auditivas para algunos usuarios.

#### Criterios de aceptación

- Pulsando en el icono del micrófono se inicia la grabación. Si existe una grabación antigua, se sobrescribe.
- Pulsando en el icono cuadrado se para la grabación.
- Pulsando en el icono del altavoz se reproduce la última grabación.

## 11. Instrucciones

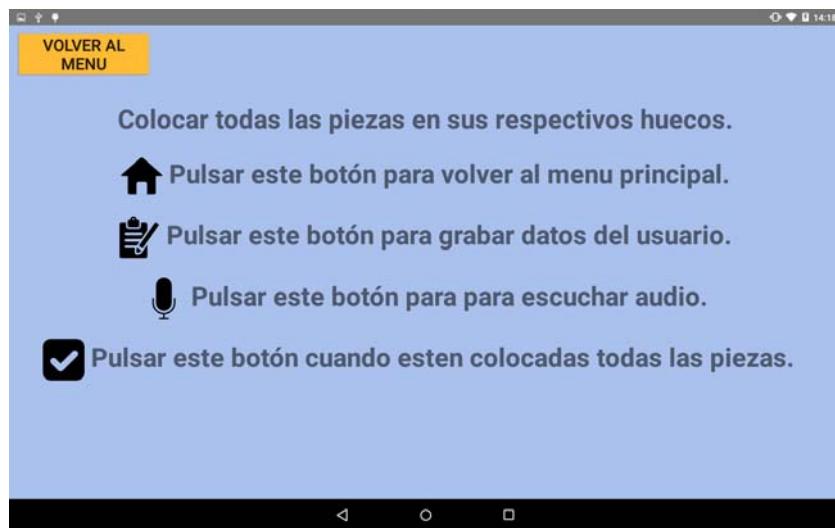


Imagen 47. Instrucciones juegos

En esta pantalla simplemente explicamos la función de los botones situados en la parte superior de los puzzles.

### Criterios de aceptación

- Al pulsar [Volver al Menú] aparece la pantalla del menú opciones.

## Diagrama de Flujo

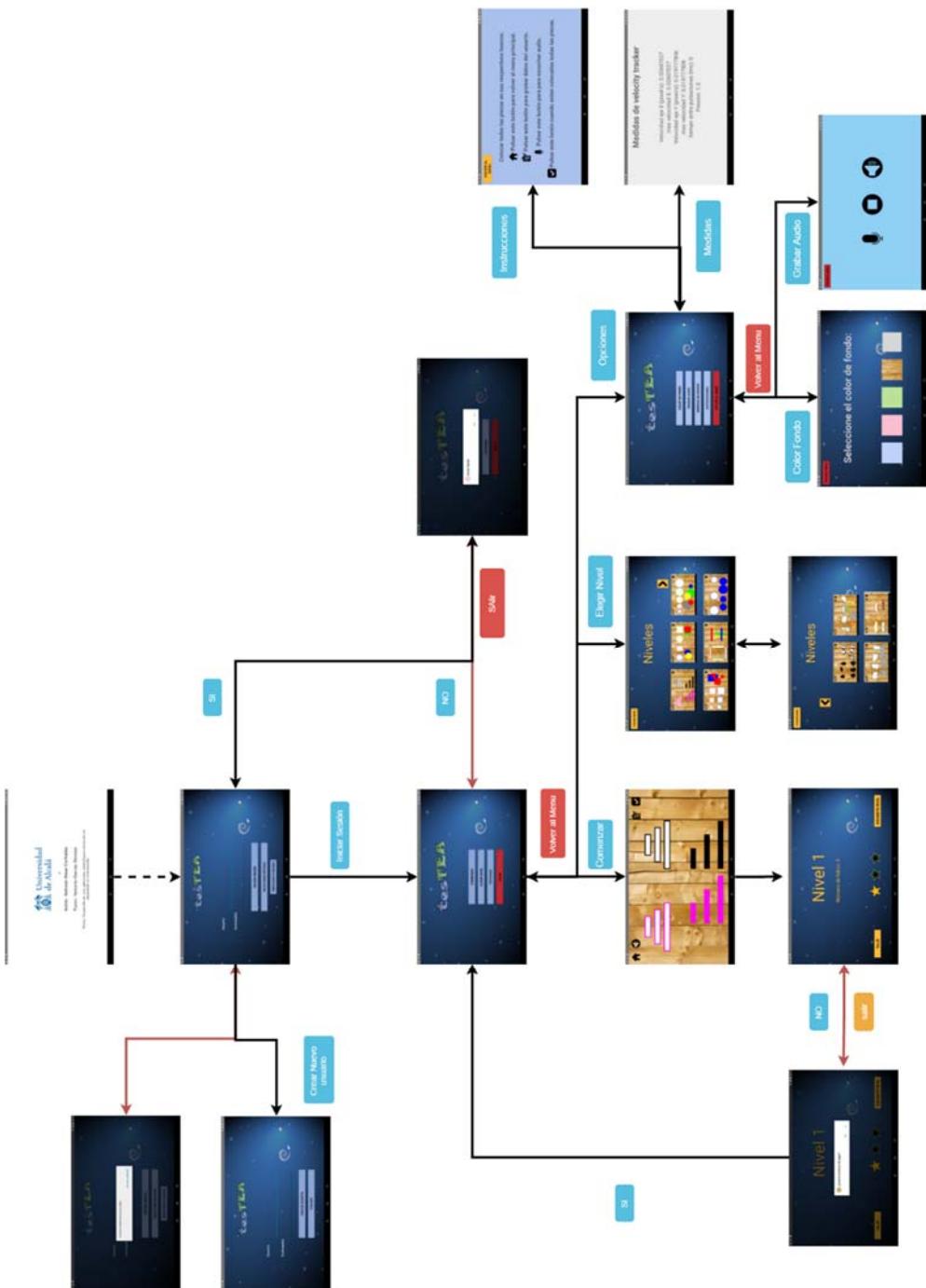


Imagen 48. Diagrama de flujo

## Aplicaciones nativas vs híbridas

Las aplicaciones móviles se dividen principalmente en dos grupos, nativas e híbridas. Las aplicaciones nativas están desarrolladas en el lenguaje específico de la plataforma. Para plataformas Android se utiliza Java y XML, en iOS se utiliza Objective C y Swift. La ventaja de desarrollar la aplicación con el lenguaje nativo de su plataforma es que podemos acceder a las APIs (Interfaz de programación de Aplicaciones) dotándolas de un mayor rendimiento.

Las aplicaciones híbridas se desarrollan con lenguajes web como Javascript, Html y CSS. Normalmente se ejecutan desde el navegador del móvil por lo tanto no pueden acceder a la mayoría de las funcionalidades hardware del dispositivo ni a las librerías propias del sistema.

La ventaja de las aplicaciones híbridas es la reutilización de código fuente para distintas plataformas, sin embargo esto conlleva peor rendimiento, experiencia de usuario y diseño. Las aplicaciones nativas se ejecutan un nivel por debajo que las híbrida, teniendo acceso al hardware y las APIs propios de cada plataforma [30].

## Diagrama de Gantt

| Actividades | fecha inicio | duración (días) | fecha fin  |
|-------------|--------------|-----------------|------------|
| Tarea 1     | 01/08/2016   | 30              | 30/08/2016 |
| Tarea 2     | 20/08/2016   | 15              | 03/09/2016 |
| Tarea 3     | 05/09/2016   | 90              | 03/12/2016 |
| Tarea 4     | 04/12/2016   | 20              | 23/12/2016 |
| Tarea 5     | 26/12/2016   | 15              | 09/01/2017 |
| Tarea 6     | 02/01/2017   | 15              | 16/01/2017 |
| Tarea 7     | 20/10/2016   | 30              | 18/11/2016 |
| Tarea 8     | 15/09/2016   | 80              | 03/12/2016 |
| Tarea 9     | 01/09/2016   | 160             | 07/02/2017 |
| Tarea 10    | 15/09/2016   | 120             | 12/01/2017 |

Tabla 2. Tareas

Tarea 1: Estudio previo Java y XML para el desarrollo de aplicaciones Android.

Tarea 2: Primera fase. Programación estática con Canvas de los puzzles de la aplicación.

Tarea 3: Segunda fase. Programación dinámica de los puzzles con método onTouch.

Tarea 4: Tercera fase. Detección de los múltiples gestos y registro de velocidades y tiempos.

Tarea 5: Cuarte fase. Login y registro de usuarios con la aplicación.

Tarea 6: Quinta fase. Definición de las variables y funciones para registrar el comportamiento del usuario.

Tarea 7: Desarrollo página web, base de datos y ficheros PHP para la comunicación entre base de datos y aplicación.

Tarea 8: Diseño y programación layouts en XML de la aplicación.

Tarea 9: Preparación documentación y memoria.

Tarea 10: Pruebas de testing sobre la aplicación.

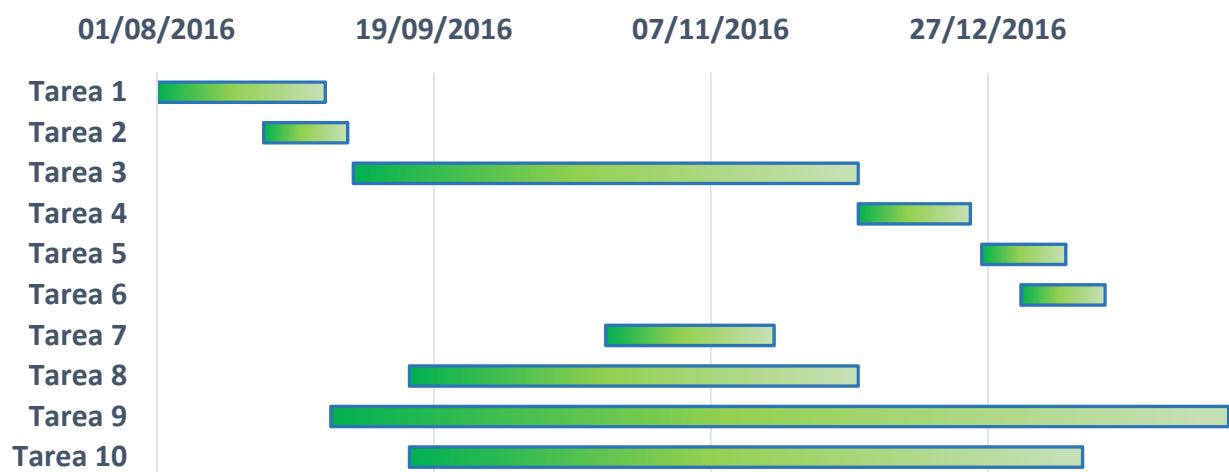


Tabla 3. Diagrama de Gantt

## Exportar base de datos con phpMyAdmin

Para exportar bases tenemos que ingresar en el administrador de bases de datos phpMyAdmin y seleccionar la base de datos sobre la que queremos realizar la exportación. A continuación seleccionamos la pestaña exportar y elegimos la opción “generar archivo descargable” y el formato que deseamos (JSON, CSV, SQL, YAML).

Imagen 49 muestra la interfaz de usuario de phpMyAdmin para exportar una tabla. La barra superior muestra la URL localhost > u878925327\_bd > datos. La barra de menú incluye Examinar, Estructura, SQL, Buscar, Insertar, Exportar (destacado), Importar, Operaciones y Seguimiento.

**Método de exportación:**

- Rápido - mostrar sólo el mínimo de opciones de configuración (seleccionado)
- Personalizado - mostrar todas las opciones de configuración posibles

**Formato:**

SQL (seleccionado)

Continuar

Imagen 49. Exportar tabla desde phpMyAdmin

## Conclusiones y futuros trabajos

El desarrollo y mercado de las aplicaciones móviles está en auge y el avance de la tecnología permite además desarrollar aplicaciones más potentes y eficientes para el usuario.

El uso de dispositivos wearables en sincronía con el móvil aumenta el control y las posibilidades de captación o registro de datos útiles del usuario. En aplicaciones mHealth, los datos registrados mediante el uso sensores, biométricos o algoritmos marcan la diferencia para poder realizar diagnósticos de mayor calidad.

Actualmente el objetivo de las aplicaciones mHealth consiste en ayudar a detectar enfermedades de manera precoz y que ayudan a prevenir riesgo de una enfermedad en el futuro. Con ayuda de periféricos o dispositivos agregados al móvil podemos simular pruebas similares a la que se realizarían en la consulta de un médico y con un software definido que sea capaz de identificar patrones característicos de enfermedades conocidas a partir de la información capturada por el hardware se puede lograr un diagnóstico fiable. De hecho actualmente ya existen aplicaciones capaces de realizar diagnósticos de prevención de enfermedades oculares a través de la cámara del móvil [31], incluso aplicaciones que son capaces de detectar VIH o sífilis mediante una fina capa de celulosa y poliéster que analiza agentes biológicos en sangre, suero y líquido peritoneal. [32]

Países en vías de desarrollo donde la posibilidad de acceder a un hospital y realizar pruebas médicas es complicada por el coste que supone, aplicaciones de este tipo pueden ayudar a mejorar y facilitar la vida de muchas personas.

El desarrollo de una aplicación nunca termina, siempre se pueden añadir nuevas funcionalidades, corregir incidencias, refactorizar el código o cambiar el diseño. Una vez realizado la aplicación en Java para Android, optando por la opción de un desarrollo nativo, se podría desarrollar la aplicación en lenguaje Swift para dispositivos con iOS.

La posibilidad de realizar futuros trabajo sobre una aplicación móvil en desarrollo es amplia. Dada la importancia de la recolección de datos para mejorar la calidad del diagnóstico, sería útil poder captar la presión del dedo sobre la pantalla, utilizar la cámara para identificar algún patrón facial que pueda resultar de utilidad o el sensor giroscópico para conocer la posición y los movimientos que hace el usuario con el dispositivo mientras juega.

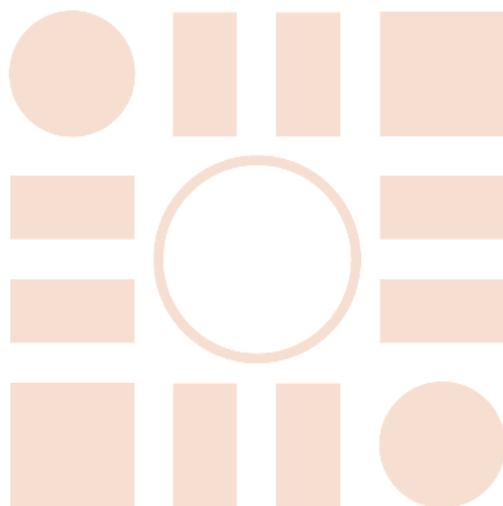
## Bibliografía

- [1] E. Scott, "App aims for faster autism diagnosis," 13 Abril 2013. [Online]. Available: <http://www.usatoday.com/story/news/nation/2013/04/13/autism-application-faster-diagnosis/2080247/>.
- [2] ImagiRation, LLC, "Mental Imgery Therapy for autism (MITA) - an early intervention computerized brain training program for children with ASD," 1 Marzo 2016. [Online]. Available: <https://clinicaltrials.gov/ct2/show/study/NCT02708290>.
- [3] G. d. t. d. l. G. p. e. m. d. p. c. t. d. e. a. e. a. primaria, "Guía de Práctica Clínica para el manejo de Pacientes con Trastornos del Espectro Autista en Atención Primaria," Madrid, Ministerio de ciencia e innovación, 2009.
- [4] A. Swartz, "Unraveling the Mystery of Autism, One Person at a Time," 8 Noviembre 2016. [Online]. Available: <http://europe.newsweek.com/unraveling-mystery-autism-one-person-time-489213?rm=eu>.
- [5] J. J. C. S. Maria Olga Escandell Bermudez, "¿Cuántas personas con autismo hay? Una revisión teórica," *International Journal of Development and Eduactional Psychology*, 2013.
- [6] INE, "Personas con alguna discapacidad que tiene diagnosticado autismo entre los 6-64 años.".
- [7] Universidad de Murcia, "Intrumetnos para la evaluación del autismo y síndrome de Asperger," [Online]. Available: <http://ocw.um.es/cc.-sociales/trastornos-del-desarrollo-y-logopedia/lectura-obligatoria-1/tema7.pdf>.
- [8] World Health Organization, "mHealth," [Online]. Available: [http://www.who.int/goe/publications/goe\\_mhealth\\_web.pdf](http://www.who.int/goe/publications/goe_mhealth_web.pdf).
- [9] iPromts, Diciembre 2015. [Online]. Available: <https://itunes.apple.com/us/app/iprompts-pro-leading-visual/id313144705?mt=8>.
- [10] K. i. creativa, "Gaido," [Online]. Available: <http://www.gaido.es/>.
- [11] "Autism Track," Septiembre 2016. [Online]. Available: <https://itunes.apple.com/us/app/autism-tracker-lite-track/id478527813?mt=8>.
- [12] "The Social Express," [Online]. Available: <http://thesocialexpress.com>.
- [13] "Stories2Learn," Mayo 2014. [Online]. Available: <https://itunes.apple.com/us/app/stories2learn/id348576875?mt=8>.

- [14] "Model me going 2 places," Septiembre 2012. [Online]. Available: <https://itunes.apple.com/us/app/model-me-going-places-2/id375669988?mt=8>.
- [15] "Cognoa," [Online]. Available: <https://itunes.apple.com/us/app/model-me-going-places-2/id375669988?mt=8>.
- [16] I. LLC, "MITA," Noviembre 2016. [Online]. Available: <https://play.google.com/store/apps/details?id=com.imagiration.mita>.
- [17] J. L. C. Candás, "Método automático y adaptativo para la detección de anomalías en la actividad física mediante un sensor de aceleración no invasivo.,," *Universidad Nacional de Educación a Distancia (España). Escuela Técnica Superior de Ingenieros Industriales. Departamento de Ingeniería Eléctrica, Electrónica y Control*, 2015.
- [18] A. Developer, "Android Manifest," [Online]. Available: <https://developer.android.com/guide/topics/manifest/manifest-intro.html>.
- [19] A. Developer, "Activity," [Online]. Available: <https://developer.android.com/reference/android/app/Activity.html>.
- [20] A. Developer, "Activity," [Online]. Available: <https://developer.android.com/reference/android/app/Activity.html>.
- [21] Android Developer, "Canvas," [Online]. Available: <https://developer.android.com/reference/android/graphics/Canvas.html>.
- [22] Android Developer, "Responding to Touch Events," [Online]. Available: <https://developer.android.com/training/graphics/opengl/touch.html>.
- [23] Android Developer, "Motion Event," [Online]. Available: <https://android.developer.com/reference/android/view/MotionEvent.html>.
- [24] Android Developer, "Handling Multi-Touch Gestures," [Online]. Available: <https://developer.android.com/training/gestures/multi.html>.
- [25] A. Developer, "Velocity Tracker," [Online]. Available: <https://developer.android.com/reference/android/view/VelocityTracker.html>.
- [26] Android Developer, "Animation Resources," [Online]. Available: <https://developer.android.com/guide/topics/resources/animation-resource.html>.
- [27] Android Developer, "MediaPlayer," [Online]. Available: <https://developer.android.com/reference/android/media/MediaPlayer.html>.
- [28] A. Developer, "AlertDialog," [Online]. Available: <https://developer.android.com/reference/android/app/AlertDialog.html>.
- [29] A. developer, "Install Android Studio," [Online]. Available: <https://developer.android.com/studio/install.html>.

- [30] C. Collado, "andro4all," 21 Enero 2016. [Online]. Available: <https://andro4all.com/2016/01/apps-nativas-vs-apps-hibridas-ventajas-desventajas>.
- [31] "Aplicación para detectar enfermedades oculares ya es realidad," 11 Julio 2015. [Online]. Available: <https://www.507noticias.com/aplicacion-para-detectar-enfermedades-oculares-ya-es-realidad/>.
- [32] E. Rivero, "Usa tu smartphone para detectar bacterias y enfermedades," 9 Abril 2015. [Online]. Available: <https://www.unocero.com/2015/04/09/usa-tu-smartphone-para-detectar-bacterias-y-enfermedades/>.
- [33] J. Cho, Desarrollo de juegos para Android, Anaya, 2016.
- [34] P. D. Stephen M. Edelson, "Tunnel vision in autism," 2016. [Online]. Available: [https://www.autism.com/advocacy\\_tunnel\\_vision](https://www.autism.com/advocacy_tunnel_vision).
- [35] B. M., "Situación y necesidades de las personas con trastornos del espectro autista en la Comunidad de Madrid," 2001. [Online]. Available: [http://www.isciii.es/htdocs/centros/enfermedadesraras/pdf/aut\\_autmad.pdf](http://www.isciii.es/htdocs/centros/enfermedadesraras/pdf/aut_autmad.pdf).
- [36] G.-V. D. Cabanyes-Truffino J, "Identificación y diagnóstico precoz de los trastornos del espectro autista," *Rev Neurol*, pp. 81-90, 2004.
- [37] P. J. Nadel S., "Early detection of autism spectrum disorders: screening between 12 and 14 months of age," *J Am Acad Nurse Pract*, pp. 408-417, 2007.
- [38] mychildwithoutlimits, "¿Cómo se diagnostica el autismo?" [Online]. Available: <http://www.mychildwithoutlimits.org/understand/autism/how-is-autism-diagnosed/?lang=es>.
- [39] R. P. Seldas, "Críterios diagnóstico DSM V," [Online].
- [40] flaticon, "flaticon," [Online]. Available: <http://www.flaticon.com/>.
- [41] A. Studio, "Gradle," [Online]. Available: <https://developer.android.com/studio/build/index.html>.
- [42] A. Developer, "Layout," [Online]. Available: <https://developer.android.com/reference/android/R.layout.html>.

Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR

