



Universidad de Valladolid

E.T.S. INGENIERÍA INFORMÁTICA
Trabajo Fin De Grado

Grado en Ingeniería en Informática

Desarrollo centrado en el usuario de un juego
"causa-efecto" sobre Android

Autor:
D. Álvaro Blanco Alonso



Universidad de Valladolid

E.T.S. INGENIERÍA INFORMÁTICA
Trabajo Fin De Grado

Grado en Ingeniería en Informática

Desarrollo centrado en el usuario de un juego
”causa-efecto” sobre Android

Autor:

D. Álvaro Blanco Alonso

Tutor:

D. Valentín Cardeñoso Payo



Este trabajo se encuentra bajo la licencia *Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 España*.

This work is licensed under the *Creative Commons Attribution-NonCommercial-ShareAlike 2.5 Spain License*.

<http://creativecommons.org/licenses/by-nc-sa/2.5/es/legalcode.es>

Resumen

El objetivo de este proyecto consiste en realizar un juego del tipo "causa-efecto" para el centro Obregón de Valladolid. Este centro se dedica a ayudar a personas con discapacidad intelectual severa. La metodología de trabajo consistirá en desarrollar un diseño centrado en el usuario.

Palabras clave proyecto, grado, universidad, android, diseño centrado en el usuario, personas con discapacidad

Abstract

The objective of this project is to make a game of cause and effect type for Obregón center in Valladolid. This center is dedicated to helping people with severe intellectual disabilities. The working methodology is to develop a user-centered design.

Keywords thesis, grade, university, android, user centered design

A mi madre, quien siempre me ha apoyado

Agradecimientos

Lo primero de todo, agradecer a mi familia su apoyo y confianza, ya que me han ayudado a conseguir todo lo que deseaba.

A mis amigos y compañeros de la universidad, por hacer de esta etapa de mi vida mas agradable.

A los profesores y personal de la universidad. Cuyo trabajo me ha dado la oportunidad de aprender tanto.

Y por ultimo a mi familia, por apoyarme siempre y estar a mi lado.

Índice general

| | |
|--|-----------|
| 1. Introducción | 1 |
| 1.1. Objetivos del proyecto | 1 |
| 1.2. Diseño Centrado en el Usuario | 2 |
| 1.2.1. Principios básicos | 3 |
| 1.3. La educación especial | 3 |
| 1.3.1. SAAC | 3 |
| 1.3.2. Trastornos motores graves | 4 |
| 1.4. El Centro Obregón | 5 |
| 1.4.1. Asprona Valladolid | 5 |
| 1.4.2. Recursos Hardware | 6 |
| 1.5. ¿Qué hace a Android Especial? | 6 |
| 1.6. Resumen de capítulos | 7 |
| 2. Gestión del proyecto | 9 |
| 2.1. Recursos | 9 |
| 2.1.1. Recursos humanos | 9 |
| 2.1.2. Recursos hardware | 10 |
| 2.1.3. Herramientas | 10 |
| 2.2. Planificación del proyecto | 11 |
| 2.3. Seguimiento | 13 |
| 2.3.1. Evolución del proyecto | 13 |
| 2.3.2. Herramientas necesitadas durante el seguimiento | 13 |
| 2.3.3. Dificultades encontradas | 13 |
| 3. Entorno Tecnológico | 15 |
| 3.1. Android | 15 |
| 3.1.1. Historia de Android | 15 |
| 3.1.2. Comparativa con otras plataformas | 16 |
| 3.1.3. Arquitectura de Android | 16 |
| 3.1.4. Versiones de Android y niveles de API | 20 |
| 3.1.5. Elección de la versión de API | 26 |
| 3.2. HTML 5 | 26 |
| 3.2.1. La historia de HTML5 | 26 |
| 3.2.2. El elemento Canvas | 27 |
| 3.3. Formatos de imagen | 28 |

| | | |
|-----------|---|-----------|
| 3.3.1. | GIF | 28 |
| 3.3.2. | JPEG | 28 |
| 3.3.3. | PNG | 29 |
| 3.3.4. | WebP | 29 |
| 3.3.5. | Conclusiones | 30 |
| 3.4. | Formatos de audio y vídeo en HTML5 | 30 |
| 3.4.1. | AVI | 31 |
| 3.4.2. | MPEG | 31 |
| 3.4.3. | MP4 H.264 | 31 |
| 3.4.4. | WebM | 32 |
| 3.5. | ¿Qué es Canvas? | 32 |
| 3.6. | Javascript | 32 |
| 3.7. | jQuery | 33 |
| 3.8. | Json | 34 |
| 3.9. | Bootstrap | 34 |
| 4. | DCU: Análisis | 37 |
| 4.1. | Análisis de usuarios | 37 |
| 4.1.1. | Usuarios objetivo | 37 |
| 4.1.2. | Perfil (grupos) de los usuarios | 38 |
| 4.1.3. | Extracción de requisitos de la IU a partir del análisis de usuarios | 39 |
| 4.1.4. | Descripción de las personas | 39 |
| 4.2. | Análisis de tareas | 40 |
| 4.3. | Modelo de dominio | 42 |
| 4.4. | Requisitos funcionales | 42 |
| 4.5. | Requisitos de usabilidad | 43 |
| 4.5.1. | Requisitos de interfaz de usuario | 43 |
| 4.6. | Especificación de casos de uso | 44 |
| 4.6.1. | Modelo de casos de uso | 44 |
| 4.6.2. | CU-1: Configurar aplicación | 45 |
| 4.6.3. | CU-2: Seleccionar minijuego | 45 |
| 4.6.4. | CU-3: Jugar a minijuego | 46 |
| 5. | DCU: Prototipo inicial | 51 |
| 5.1. | Introducción | 51 |
| 5.1.1. | ¿Qué es un prototipo? | 51 |
| 5.1.2. | Tipos de prototipos | 51 |
| 5.2. | Entrevistas con usuarios | 52 |
| 5.3. | Evaluación prototipo baja definición | 53 |
| 5.3.1. | Menú principal | 53 |
| 5.3.2. | Juego de los fuegos artificiales | 54 |
| 5.3.3. | Juego de las partículas | 56 |
| 5.3.4. | Juego de las estrellas | 57 |
| 5.3.5. | Cantantes | 59 |
| 5.3.6. | Juego de los trazos | 60 |
| 5.3.7. | Juego del huevo | 62 |
| 5.3.8. | Dibujos animados | 64 |
| 5.3.9. | Pictogramas y canciones | 65 |

| | |
|--|------------|
| 5.3.10. Comentarios adicionales realizados por los evaluados | 66 |
| 5.3.11. Conclusiones evaluación prototipo baja definición | 67 |
| 6. DCU: Diseño | 69 |
| 6.1. El modelo de capas | 69 |
| 6.2. El patrón MVC | 69 |
| 6.3. Elección del framework | 71 |
| 6.4. Idea del juego | 73 |
| 6.5. Canvas frente a Flash | 74 |
| 6.6. Diseño de la aplicación | 75 |
| 6.6.1. Definir estilo | 76 |
| 6.6.2. Diseño de contenidos | 79 |
| 6.6.3. Diseño de persistencia | 81 |
| 7. DCU: Implementación y Pruebas | 85 |
| 7.1. Preparación del proyecto | 85 |
| 7.2. Implementación del prototipo funcional | 85 |
| 7.2.1. Organización de las carpetas | 85 |
| 7.2.2. Bootstrap | 87 |
| 7.2.3. Fuente | 87 |
| 7.2.4. Iconos (glyphicons) | 88 |
| 7.2.5. Thumbnails) | 88 |
| 7.2.6. Animaciones) | 88 |
| 7.3. Integración con Phonegap | 89 |
| 7.4. Validación de la aplicación | 90 |
| 7.4.1. Metodología | 90 |
| 7.4.2. Pruebas de verificación | 91 |
| 7.4.3. Resultados de las pruebas | 97 |
| 7.5. Conclusiones | 97 |
| 8. Conclusiones y trabajos futuros | 99 |
| 8.1. Contribuciones | 99 |
| 8.2. Mejoras y trabajos futuros | 99 |
| A. Definición de términos y abreviaturas | 101 |
| B. Contenido del CD-ROM | 103 |
| Bibliografía | 105 |

Índice de figuras

| | |
|--|----|
| 2.1. Ciclo vida diseño centrado usuario | 12 |
| 3.1. Arquitectura de Android | 18 |
| 3.2. Dispositivos Android según plataforma instalada | 26 |
| 3.3. Representación del DOM | 35 |
| 4.1. Modelo de dominio | 42 |
| 4.2. Diagrama de casos de uso | 44 |
| 5.1. Prototipo Low: Menú Principal | 53 |
| 5.2. Prototipo Low: Fuegos Artificiales | 54 |
| 5.3. Prototipo Low: Fuegos Artificiales | 55 |
| 5.4. Prototipo Low: Fuegos Artificiales | 55 |
| 5.5. Prototipo Low: Juego de las partículas | 56 |
| 5.6. Prototipo Low: Juego de las partículas | 56 |
| 5.7. Prototipo Low: Juego de las partículas | 57 |
| 5.8. Prototipo Low: Juego de las estrellas | 57 |
| 5.9. Prototipo Low: Juego de las estrellas | 58 |
| 5.10. Prototipo Low: Juego de las estrellas | 58 |
| 5.11. Prototipo Low: Cantantes | 59 |
| 5.12. Prototipo Low: Cantantes | 59 |
| 5.13. Prototipo Low: Juego de los trazos | 60 |
| 5.14. Prototipo Low: Juego de los trazos | 61 |
| 5.15. Prototipo Low: Juego de los trazos | 61 |
| 5.16. Prototipo Low: Juego de los trazos | 62 |
| 5.17. Prototipo Low: Juego del huevo | 62 |
| 5.18. Prototipo Low: Juego del huevo | 63 |
| 5.19. Prototipo Low: Juego del huevo | 63 |
| 5.20. Prototipo Low: Juego del huevo | 64 |
| 5.21. Prototipo Low: Dibujos animados | 64 |
| 5.22. Prototipo Low: Dibujos animados | 65 |
| 5.23. Prototipo Low: Dibujos animados | 65 |
| 5.24. Prototipo Low: Pictogramas y canciones | 66 |
| 5.25. Prototipo Low: Pictogramas y canciones | 66 |
| 5.26. Prototipo Low: Pictogramas y canciones | 67 |

| | |
|---|----|
| 6.1. Diagrama del patrón MVC | 70 |
| 6.2. PhoneGap | 71 |
| 6.3. Funcionamiento de PhoneGap | 72 |
| 6.4. Idea del juego | 74 |
| 6.5. Arquitectura de PhoneGap | 75 |
| 6.6. Estructura de la aplicación web | 76 |
| 6.7. Diseño estructural menú principal | 78 |
| 6.8. Diseño estructural minijuegos "Pictogramas", "Cantantes" y "Dibujos" | 78 |
| 6.9. Diseño estructura del resto de ventanas de la aplicación | 79 |
| 6.10. Página web programación infantil RTVE | 80 |
| 6.11. Mapa mental persistencia aplicación | 82 |
| 7.1. Creación del proyecto | 85 |
| 7.2. Creación del proyecto | 86 |
| 7.3. Organización carpetas HTML | 86 |
| 7.4. Organización de la carpeta assets | 87 |
| 7.5. Estructura de Bootstrap | 88 |
| 7.6. Instalación Phonegap paso 1 | 89 |
| 7.7. Instalación Phonegap paso 2 | 89 |

Índice de tablas

| | |
|--|----|
| 3.1. Cuota mercado principales plataformas moviles | 16 |
| 3.2. Comparativa de las principales plataformas moviles | 17 |
| 3.3. Versiones kernel Linux Android AOSP | 19 |
| 3.4. Soporte canvas navegadores | 28 |
| 3.5. Soporte imagenes navegadores | 30 |
| 4.1. Perfil general de los alumnos | 38 |
| 4.2. Perfil general de los educadores | 38 |
| 4.3. Grupos de usuarios | 39 |
| 4.4. Extraccion de requisitos de la IU | 40 |
| 4.5. CU-1: Configurar aplicación | 45 |
| 4.6. CU-2: Seleccionar minijuego | 45 |
| 4.7. CU-3: Jugar a minijuego | 46 |
| 4.8. CU-3.1: Fuegos artificiales | 46 |
| 4.9. CU-3.2: Partículas | 47 |
| 4.10. CU-3.3: Estrellas | 47 |
| 4.11. CU-3.4: Trazos | 48 |
| 4.12. CU-3.5: Cantantes | 48 |
| 4.13. CU-3.6: Huevo | 49 |
| 4.14. CU-3.7: Dibujos | 49 |
| 4.15. CU-3.8: Pictogramas | 50 |
| 6.1. Colores de fondo | 77 |
| 7.1. Prueba de selección de minijuegos | 91 |
| 7.2. Prueba de carga de imágenes del menú principal | 91 |
| 7.3. Prueba de funcionamiento del minijuego Estrellas | 92 |
| 7.4. Prueba de funcionamiento del minijuego Huevo | 92 |
| 7.5. Prueba de funcionamiento del minijuego Dibujos | 93 |
| 7.6. Prueba de carga del minijuego Dibujos | 93 |
| 7.7. Prueba de funcionamiento del minijuego Pictogramas | 94 |
| 7.8. Prueba de carga del minijuego Pictogramas | 94 |
| 7.9. Prueba de funcionamiento del minijuego Cantantes | 95 |
| 7.10. Prueba de carga del minijuego Cantantes | 95 |
| 7.11. Prueba de funcionamiento del minijuego Fuegos artificiales | 96 |

| | |
|---|----|
| 7.12. Prueba de funcionamiento del minijuego Trazos | 96 |
| 7.13. Prueba de funcionamiento del minijuego Particulas | 97 |
| 7.14. Resultados de las pruebas | 98 |

Introducción

Los avances tecnológicos están teniendo un gran impacto en nuestras vidas. Por ejemplo están consiguiendo que las personas con discapacidad accedan a la información e interactúen de una forma que no tiene precedentes. Les da la oportunidad de participar en la sociedad de una forma que de otro modo sería imposible. Con aplicaciones y webs accesibles, las personas con discapacidad pueden hacer más cosas por sí mismas sin tener que depender de los demás. Las personas ciegas y también las personas con discapacidad cognitiva, que tienen problemas para procesar la información escrita, pueden leer el periódico (a través de lectores de pantalla que leen el texto en voz alta desde el ordenador). Las personas sordas pueden acceder a noticias de máxima actualidad, lo mismo que las personas sordomudas (a través de dispositivos dinámicos de Braille), mientras que antes sólo accedían aquellos que podían oír la radio o la televisión.

1.1. Objetivos del proyecto

El objetivo de este proyecto es utilizar la metodología de diseño centrado en el usuario (DCU) en el desarrollo de un juego "causa-efecto" para sistemas Android.

Funciones del producto

- Apretar fuegos artificiales y cohetes, cuando el niño los toque explota
- Pulsar un dedo y ver partículas que vuelan zumbando desde la punta del dedo
- Pantalla con estrellas que al tocarlas se iluminan y las puede desplazar con el dedo
- Pasar el dedo por la pantalla y crear trazos de colores
- Tocar la foto de un cantante y reproducir un vídeo de su música
- En la pantalla aparece un huevo, al tocarlo se resquebraja haciendo ruido y sale un pollito
- Aparece un personaje de dibujos animados y al pulsarlo enlaza con un episodio
- Aparece un pictograma referido a una canción y al pulsar se oye o ve el vídeo de la canción

1.2. Diseño Centrado en el Usuario

El Diseño Centrado en el Usuario (DCU), o User Centered Design (UCD), es definido por la Usability Professionals Association (UPA) como un enfoque de diseño cuyo proceso está dirigido por información sobre las personas que van a hacer uso del producto. El Informe APEI sobre usabilidad [31] comenta que el origen de esta visión se enmarca en el diseño industrial y militar de la década de los cincuenta. Por entonces, los diseñadores estaban convencidos de que la optimización y adaptación al ser humano del diseño de productos respondía a un minucioso proceso de investigación en antropometría,ergonomía, arquitectura o biomecánica.

Henry Dreyfuss, autor del libro *Designing for people* (1955) popularizó la concepción del "diseño como proceso" a partir de sus diseños de teléfonos de la serie 500 para Bell Telephones. Este diseñador industrial, pionero del diseño centrado en el usuario, estudió cómo se construían los teléfonos, cómo se percibían y eran utilizados por las personas [30]. En esa misma década los avances informáticos daban luz a la segunda generación de ordenadores, pero la atención de los ingenieros de software seguía centrada en comprender la forma de trabajo del ordenador, y muy poco en comprender la forma de trabajo de las personas.

Podría parecer un descuido, pero lo cierto es que el cambio que se produjo con la incorporación de los transistores en los circuitos de los ordenadores no ayudó a dirigir la atención hacia los nuevos usuarios. Estos transistores, además de ser escasos, tenían serias limitaciones de procesamiento y almacenamiento, que requerían una dedicación mayor por parte de los ingenieros e informáticos y que, en consecuencia, relegaban el interés por el comportamiento o las características de aquellas personas que comenzaban a experimentar procesos de interacción con las máquinas.

En la década de los ochenta comienza la plena expansión del diseño centrado en el usuario, como atestigua el aumento de revistas, artículos y foros especializados en los estudios de Interacción Persona-Computador (IPC) o Human-Computer Interaction (HCI).

Norman [34], profesor de la Northwestern University y cofundador de Nielsen Norman Group, fue quien comenzó a utilizar el termino User Centered System Design en el conjunto de conferencias presentadas por su equipo, en la primera CHI Conference (1983), organizada por la ACM Special Interest Group on Computer-Human Interaction (SIGCHI), en Boston (Massachusetts, USA).

El concepto de DCU se utilizó como marco de trabajo, investigación y desarrollo de principios del diseño de interfaces de usuario. Era el momento de observar cómo la gente usaba los sistemas y creaba sus propios modelos mentales a partir de los procesos de interacción. Tres fueron los términos que debían ser valorados para entender estos procesos:

- El modelo conceptual: ofrecido por el diseñador del sistema.
- Interfaz: la imagen que el sistema presenta al usuario.
- El modelo mental: desarrollado por el usuario a partir de la imagen.

Como señala Norman [34], la imagen del sistema guía al usuario en la adquisición y construcción de un modelo mental ajustado al modelo conceptual creado por el diseñador.

De este modo, el enfoque del DCU persigue asegurar la consecución de un producto con la funcionalidad adecuada para usuarios concretos. El objetivo de esta filosofía es ofrecer respuesta a preguntas como ¿quién usará este sistema?, ¿qué es lo que va a hacer con él? ó ¿qué información necesitará para alcanzar sus objetivos?

Se habla del DCU como una filosofía o un enfoque porque como diseñadores partimos de una premisa que condicionará todas nuestras acciones: el usuario debe ubicarse en el centro de toda decisión de diseño. No sólo diseñamos productos, diseñamos experiencias de usuario, porque no es

posible entender el producto desvinculado de su uso, su contexto, o de las necesidades y motivaciones del usuario final.

El objetivo final del DCU es, por tanto, lograr la satisfacción de las necesidades de todos sus usuarios potenciales, adaptar la tecnología utilizada a sus expectativas y crear interfaces que faciliten la consecución de sus objetivos.

1.2.1. Principios básicos

Gould en su artículo "Designing for Usability: Key Principles and What Designers Think" [24] menciona que el diseño centrado en el usuario se caracteriza por los siguientes principios de diseño.

- **Hacer foco temprano en los usuarios y sus tareas:** El objetivo es entender las características cognitivas, de comportamiento y las actitudes de los usuarios, las tareas que realizan, de qué forma y en qué ambiente lo hacen. Este entendimiento se logra a través del contacto directo con los usuarios representativos y mediante otras fuentes indirectas. Si se desea que el usuario comprenda los productos desarrollados, primero debemos comprender al usuario y sus necesidades reales.
- **Participación activa de los usuarios:** Los usuarios representativos deben participar activa y constantemente a lo largo de todo el proceso de desarrollo.
- **Realizar mediciones empíricas:** El diseño funciona sólo si los usuarios reales deciden que funciona. Los usuarios deben hacer pruebas en simuladores o prototipos y sus reacciones y forma de interacción deben ser observadas, grabadas y analizadas. Incorporar pruebas de usabilidad a lo largo del proceso de desarrollo brinda a los usuarios la oportunidad de dar feedback sobre el diseño antes de que el producto se ponga en marcha.
- **Actividades de diseño integradas:** El proceso de desarrollo debe contemplar actividades de diseño. El diseño interno del producto debe ser consistente con las necesidades de las interfaces de usuario. El diseño debe ser representado en una forma que resulte fácilmente entendible para los usuarios y otros interesados en el sistema (stakeholders).
- **Prototipar y evaluar iterativamente:** los prototipos deben hacerse continuamente desde etapas tempranas y deben usarse para visualizar y evaluar ideas con los usuarios. El ciclo de prototipar-testear-prototipar debe realizarse hasta que los objetivos de usabilidad definidos sean alcanzados.
- **Filosofía de DCU:** El DCU requiere que todos los miembros del proyecto tengan una actitud centrada en el usuario. Expertos en usabilidad deben ser involucrados desde el principio al fin del desarrollo. El proceso debe ser dirigido por equipos multidisciplinarios.

1.3. La educación especial

1.3.1. Sistemas Aumentativos y Alternativos de Comunicación

Los Sistemas Aumentativos y Alternativos de Comunicación (SAAC) son formas de expresión distintas al lenguaje hablado, que tienen como objetivo aumentar (aumentativos) y/o compensar (alternativos) las dificultades de comunicación y lenguaje de muchas personas con discapacidad.

La comunicación y el lenguaje son esenciales para todo ser humano, para relacionarse con los demás, para aprender, para disfrutar y para participar en la sociedad y hoy en día, gracias a estos sistemas, no deben verse frenados a causa de las dificultades en el lenguaje oral. Por esta razón, todas las personas, ya sean niños, jóvenes, adultos o ancianos, que por cualquier causa no han adquirido o han perdido un nivel de habla suficiente para comunicarse de forma satisfactoria, necesitan usar un SAAC.

Entre las causas que pueden hacer necesario el uso de un SAAC encontramos la parálisis cerebral (PC), la discapacidad intelectual, los trastornos del espectro autista (TEA), las enfermedades neurológicas tales como la esclerosis lateral amiotrófica (ELA), la esclerosis múltiple (EM) o el párkinson, las distrofias musculares, los traumatismos cráneo-encefálicos, las afasias o las pluridis-capacidades de tipologías diversas, entre muchas otras.

La Comunicación Aumentativa y Alternativa (CAA) no es incompatible sino complementaria a la rehabilitación del habla natural, y además puede ayudar al éxito de la misma cuando éste es posible. No debe pues dudarse en introducirla a edades tempranas, tan pronto como se observan dificultades en el desarrollo del lenguaje oral, o poco después de que cualquier accidente o enfermedad haya provocado su deterioro. No existe ninguna evidencia de que el uso de CAA inhiba o interfiera en el desarrollo o la recuperación del habla.

1.3.2. Trastornos motores graves

Parálisis cerebral infantil

Es una afección orgánica del encéfalo que se puede producir en el feto por falta de oxígeno o por infecciones de la madre y en el neonato (durante el nacimiento) por falta de oxigenación (anoxia), por afecciones metabólicas e, incluso, por accidentes. Puede tener estos síntomas: rigidez y excitabilidad en brazos y piernas, movimientos anormales e involuntarios, rigidez muscular, . . . Según los miembros afectados esta enfermedad se denomina: tetraplejía (afecta a brazos y piernas), paraplejía (afecta a las piernas), hemiplejía (afecta a los miembros de un lado del cuerpo), monoplejía (afecta a un solo miembro).

Espina bífida

Es una enfermedad producida por el desarrollo incompleto de las vertebrae durante el embarazo que deja un hueco a la altura de la región lumbar y produce una malformación de la medula espinal. Se considera que el defecto se produce por la interacción de factores genéticos con un agente ambiental desconocido. Los problemas más importantes son: pérdida de sensibilidad de la piel más abajo del nivel de la lesión de la medula; debilidad más abajo del nivel de la lesión de la medula que puede provocar incluso una parálisis de los miembros inferiores; debilidad de los músculos de la vejiga y del intestino, lo que puede provocar incontinencia urinaria e intestinal.

Distrofia muscular progresiva

La causa de esta grave alteración puede ser genética o de origen desconocido. Consiste en la progresiva pérdida de fuerza muscular provocada por la degeneración de las fibras musculares. Suele afectar primero a los músculos de las piernas y se puede llegar a extender a los músculos del resto del cuerpo.

Poliomielitis o parálisis infantil

Es una enfermedad infecciosa de etiología vírica que ataca a neuronas motoras, lesionando porciones de fibras nerviosas de la medula espinal. Puede provocar deformaciones óseas y parálisis.

Trastornos de tics

Es un movimiento motor súbito, rápido, no rítmico y estereotipado. Asociados a este tipo de trastorno aparecen:

- **Movimientos coreiformes:** movimientos oscilantes de carácter fortuito, no repetitivo e irregulares.
- **Movimientos espasmódicos:** Son movimientos lentos, muy estereotipados y de mayor implicación muscular que un tic normal.
- **Sincinesias:** Son movimientos involuntarios asociados a otros movimientos voluntarios, como por ejemplo, cortar con tijeras y al mismo tiempo hacer muecas con la boca.

1.4. El Centro Obregón

El Centro Educativo y Asistencial Obregón tiene como usuarios a personas con discapacidad intelectual, asociada en la mayoría de los casos a una discapacidad motora grave y en menor número a discapacidades visuales y/o auditivas. Son personas, por lo tanto, que precisan de apoyos extensos y generalizados. Tiene como objetivos en las distintas etapas los siguientes: en la etapa Infantil la estimular y desarrollar las capacidades de los alumnos en todos los ámbitos de la experiencia. En la Etapa Básica promover el máximo desarrollo de capacidades y habilidades de los alumnos para que les permita participar en los diferentes entornos y actividades que podrán encontrar en el transcurso de su vida. El objetivo fundamental de la Transición a la vida Adulta es afianzar y continuar, en su caso, la formación ya iniciada y facilitar el paso de la etapa escolar de los alumnos al mundo ocupacional o laboral...

En todas las etapas se promueve el máximo grado de calidad de vida con especialmente incidencia en las vertientes de salud y bienestar. El centro Obregón pertenece Asprona Valladolid

1.4.1. Asprona Valladolid

ASPRONA es una Asociación sin ánimo de lucro (ONL) declarada de interés social y utilidad pública que se constituye como asociación para la defensa de los derechos de las personas con discapacidad intelectual y de sus familias en Valladolid y provincia.

La organización inició su andadura en 1962 como asociación de padres de personas con discapacidad intelectual. A lo largo de estos años ha ido desarrollando una amplia red de centros y servicios para satisfacer las necesidades del colectivo.

Dada su experiencia y trayectoria, puede ser considerada como la organización más representativa del sector de atención a personas con discapacidad intelectual en Valladolid.

En el ámbito nacional, ASPRONA forma parte de FEAPS (Confederación Nacional de Asociaciones que prestan servicio a personas con discapacidad intelectual). Este movimiento aglutina el esfuerzo de las organizaciones del sector promoviendo iniciativas innovadoras, eficaces y de calidad en la forma de organizar y prestar servicios.

1.4.2. Recursos Hardware

El cliente dispone actualmente de 8 tabletas Android.

- 6 Tabletas Archos con Android 2.3.1
- 2 Tabletas Samsung Galaxy Tab 2 10.1” con Android 4.0.3 ¹

1.5. ¿Qué hace a Android Especial?

Existen muchas plataformas para móviles, sin embargo Android presenta una serie de características que lo hacen diferente. Es el primero que combina en una misma solución las siguientes cualidades:

- Plataforma realmente abierta. Es una plataforma de desarrollo libre basada en Linux y de código abierto. Una de sus grandes ventajas es que se puede usar y personalizar el sistema sin pagar regalías a ninguna empresa.
- Portabilidad asegurada. Las aplicaciones finales son desarrolladas en Java, lo que nos asegura que podrán ser ejecutadas en gran variedad de dispositivos, tanto presentes como futuros. Esto se consigue gracias al concepto de máquina virtual.
- Arquitectura basada en componentes inspirados en Internet. Por ejemplo, el diseño de la interfaz de usuario se hace en xml, lo que permite que una misma aplicación se ejecuta en un móvil de pantalla reducida, en una tablet o en un netbook.
- Filosofía de dispositivo siempre conectado a Internet.
- Gran cantidad de servicios incorporados: por ejemplo, localización basada tanto en GPS como en torres de telefonía móvil. Incorpora potentes bases de datos con SQL. Reconocimiento y síntesis de voz, navegador, mapas ...
- Alto nivel de seguridad. Los programas se encuentran aislados unos de otros gracias al concepto de ejecución dentro de una caja que incorpora la máquina virtual. Cada aplicación dispone de una serie de permisos que limitan su rango de actuación (servicios de localización, acceso a Internet...)
- Optimización para baja potencia y poca memoria. Por ejemplo, Android utiliza la Máquina Virtual Dalvik. Se trata de una implementación de Google de la máquina virtual de Java optimizada para dispositivos móviles.
- Alta calidad de gráficos y sonido: gráficos vectoriales suavizados, animaciones inspiradas en Flash, gráficos en 3 dimensiones basados en OpenGL. Incorpora codecs estándar mas comunes de audio y vídeo incluyendo H.264 (AVC), MP3, AAC, ...

Con todo esto en cuenta, Android combina características muy interesantes. También hay que tener en cuenta que Android se esta convirtiendo en el nuevo estándar de sistema operativo para móviles, al menos en el caso de sistemas operativos libres.

En conclusión Android ofrece una forma sencilla y novedosa de implementar potentes aplicaciones para móviles.

¹En noviembre de 2013 las Galaxy Tab 2 10.1” recibieron una actualización a Android 4.2.2

1.6. Resumen de capítulos

A continuación se resume brevemente cada uno de los capítulos que forman esta memoria para tener una idea general de como se ha organizado este trabajo de acuerdo a los objetivos planteados en el apartado anterior.

- **Capítulo 2: Gestión del proyecto**

En este capítulo realizamos una planificación del proyecto. También realizamos una descripción de los recursos y herramientas utilizadas para elaborar tanto la aplicación en sí como de su documentación.

- **Capítulo 3: Entorno Tecnológico**

En este capítulo se describe las tecnologías utilizadas en la realización del proyecto, para así tener una idea global de su funcionamiento.

- **Capítulo 4: DCU: Análisis**

En este capítulo se realiza un análisis centrado en el usuario y se especifican los casos de uso y requisitos del proyecto.

- **Capítulo 5 DCU: Prototipo inicial**

En este capítulo se muestra el prototipo no funcional realizado así como de su evaluación por parte del cliente.

- **Capítulo 6: DCU: Diseño**

En este importante capítulo, se diseña la solución mas adecuada para el proyecto.

- **Capítulo 7: DCU: Implementación y Pruebas**

En este capítulo se especifican los pasos que se han tenido que dar para construir el prototipo funcional. Así mismo también se muestran los resultados obtenidos de las pruebas realizadas y se comprueba si la aplicación cumple con todos los requisitos expuestos.

- **Capítulo 8: Conclusiones y trabajos futuros**

En este capítulo indicamos las conclusiones a las que hemos llegado durante la realización de este trabajo de fin de grado. Así mismo detallamos las acciones que habría que realizar en el caso de que alguien quisiera seguir desarrollando la aplicación.

- **Apéndice A: Definición de términos y abreviaturas**

En este primer apéndice se trata de dar explicación a los términos y abreviaturas utilizadas.

- **Apéndice B: Contenido del CD-ROM**

En el ultimo apéndice se trata sobre el contenido del CD-ROM que se adjunta a este proyecto.

Gestión del proyecto

En este capítulo hacemos una planificación temporal y monetaria del proyecto, así como un seguimiento del mismo con el objetivo de calcular los costes tanto en tiempo como en dinero de la aplicación. También hacemos una descripción de los recursos y las herramientas utilizadas tanto para elaborar la aplicación en sí, como de su documentación.

2.1. Recursos

Los recursos son todos aquellos elementos tanto humanos como materiales que son utilizados en la elaboración del proyecto.

2.1.1. Recursos humanos

Los recursos humanos indican los roles y responsabilidades que serán necesarios en la realización del proyecto. Al tratarse de un proyecto Fin de Carrera, todos los roles recaen exclusivamente sobre una persona, el autor del proyecto, es decir, Álvaro Blanco Alonso. Los distintos roles y responsabilidades que son necesarios en este proyecto son los siguientes:

- **Jefe de proyecto:** Encargado de gestionar el proyecto, es decir, se encarga de la organización, planificación y seguimiento del mismo.
- **Analista:** Se encarga de obtener la especificación de requisitos de usuario, establecer los casos de uso, redactar los requisitos software y modelar los procesos y tareas a implementar.
- **Diseñador:** Se encarga de obtener los casos de uso reales, las distintas interfaces de la aplicación, además de definir la arquitectura del sistema a un nivel más bajo.
- **Programador:** Codifica el sistema partiendo del modelo de procesos y las directrices fijadas por el diseñador. También es el responsable de implantar el sistema debidamente. Además, se encarga también de la construcción del manual de usuario.
- **Probador:** Se encarga de efectuar las pruebas de funcionamiento, de interfaz de usuario, de carga, de accesibilidad y de seguridad del sistema.

2.1.2. Recursos hardware

Los recursos hardware son aquellos elementos físicos que han sido necesarios en el desarrollo de este proyecto. Los recursos hardware utilizados son los siguientes:

- **Ordenador portátil:** Asus A55V. Intel Core i7 4Gb de RAM, 750Gb HDD, Nvidia Geforce 610M.
- **Tableta Android:** Samsung Galaxy Tab 2 con sistema operativo Android 4.0.3 (Ice cream Sandwich)¹

2.1.3. Herramientas

Las herramientas a utilizar para la confección de este trabajo, tanto documentación como desarrollo software son las siguientes:

Herramientas para la documentación

- **Texmaker 3.5.2:** Herramienta para escribir documentos de texto, multiplataforma, que integra muchas herramientas necesarias para desarrollar documentos con LaTeX, en una sola aplicación.
- **JabRef 2.9.2:** Software de gestión bibliográfica gratuito y multiplataforma que utiliza BibTeX como formato nativo. JabRef proporciona una interfaz fácil de usar para la edición de archivos de tipo BibTeX, para la importación de datos de bases de datos científicos en línea, y para la gestión y la búsqueda de archivos BibTeX.[45]
- **StarUML 5.0.2:** Herramienta para el modelado UML de código abierto y gratuita.

Herramientas para la implementación

- **Notepad++:** Es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación.
- **Pencil:** Es una herramienta de prototipado de interfaces gráficas de usuario de código abierto.
- **XnConvert:** XnConvert fue creado por los responsables de XnView, un popular visor de imágenes. Se trata de un procesador de imágenes por lotes para Windows, Mac y Linux compatible con más de 500 formatos diferentes. Entre otras cosas, permite retocar aspectos de la imagen como el brillo o la saturación, cambiar el tamaño, editar los metadatos e incluso aplicar filtros y efectos a todo un paquete de fotografías de una sola vez.
- **A5 HTML5 Animator:** Es una solución de software integral para la creación de animaciones web basadas en el nuevo estándar Html 5. Este software permite crear contenido interactivo sin necesidad de conocimientos de programación.

El software incluye un editor WYSIWYG para la creación y posicionamiento de elementos, tales como textos, imágenes o videos. Y las animaciones se realizan con la ayuda de fotogramas, insertados tanto manual como automáticamente.

¹Posteriormente sistema operativo actualizado a Android 4.2.2 (Jelly Bean)

- **Eclipse:** Entorno de Desarrollo Integrado (IDE, Integrated Development Environment) abierto, gratuito y extensible. Cuenta con un editor de código, un compilador/intérprete y un depurador. Eclipse sirve como IDE Java y cuenta con numerosas herramientas de desarrollo de software. También da soporte a otros lenguajes de programación, como son C/C++, Cobol, Fortran, PHP o Python. A la plataforma base de Eclipse se le pueden añadir extensiones (plugins) para extender la funcionalidad.
- **Plugin de Aptana Studio 3 para Eclipse:** Aptana es un IDE basado en Eclipse y desarrollado por Aptana, Inc. Provee las herramientas para el desarrollo de software en lenguajes como: Php, Python, Ruby, CSS, Ajax, HTML y Adobe AIR. Puede ser instalado como programa independiente o como plugin para Eclipse.
- **FreeMind:** Herramienta de código abierto que permite la elaboración y manipulación de mapas mentales o conceptuales.

2.2. Planificación del proyecto

El proceso de un Diseño Centrado en el Usuario como el propuesto en este trabajo se divide en varias fases o etapas, algunas de las cuales tienen carácter iterativo. Sirva como aproximación la figura 2.1, donde podemos observar las fases en las que se divide[27].

Las distintas fases son las siguientes:

- **Planificación:** En esta etapa se identifican los objetivos del sitio, así como las necesidades, requerimientos y objetivos del cliente.
- **Diseño:** La etapa de Diseño es el momento del proceso de desarrollo para la toma de decisiones acerca de cómo diseñar o rediseñar, en base siempre al conocimiento obtenido en la etapa de planificación, así como a los problemas de usabilidad descubiertos en etapas de prototipado y evaluación.
- **Prototipado:** La etapa de prototipado se basa en la elaboración de modelos o prototipos de la interfaz del sitio. Su aspecto no se corresponde exactamente con el que tendrá el sitio una vez finalizado, pero pueden servir para evaluar la usabilidad del sitio sin necesidad de esperar a su implementación.
- **Evaluación:** La evaluación de la usabilidad es la etapa más importante en el proceso de Diseño Centrado en el Usuario, y con ella nos permite obtener información sobre el diseño de nuestro software y de esta manera mejorarlo y/o validarlo.
- **Implementación y lanzamiento:** La implementación no es más que la creación de los elementos que van a formar parte de la aplicación una vez que se encuentre en producción. Una vez implementado el sitio y testada su funcionalidad se procede al lanzamiento de la aplicación, que consiste en su puesta a disposición para los usuarios. Se trata de un evento importante, muchas veces erróneamente apresurado debido a la necesidad de cumplir plazos de entrega.
- **Mantenimiento y seguimiento:** Un software no es un ente estático, y hay que estar pendiente de él durante todo el ciclo de vida del producto, proporcionando actualizaciones en forma de aumentos de funcionalidad y corrección de errores.

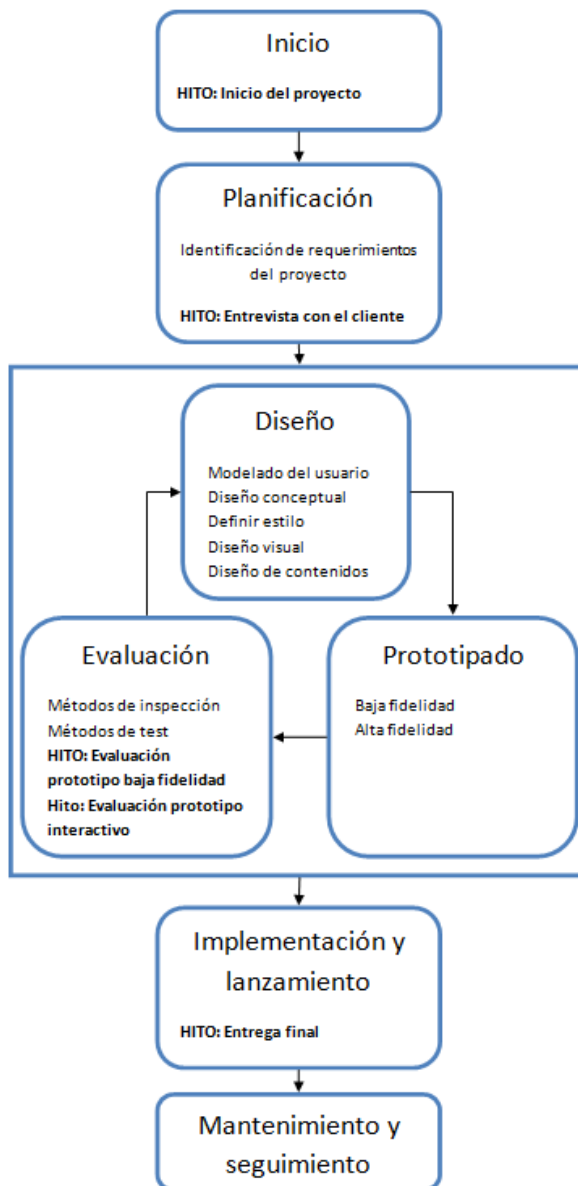


Figura 2.1: Ciclo vida de un diseño centrado en el usuario

Este proyecto también cuenta con una serie de hitos distribuidos en tiempo. Los hitos (milestones) son las etapas completadas dentro del plan de desarrollo de software que se usan para medir el avance del proyecto.

Los hitos de este proyecto son los siguientes:

- **Inicio:** Marca el inicio del proyecto.

- **Entrevista con el cliente:** Este hito consiste en realizar una primera visita al cliente de nuestra aplicación, para así hacernos una idea del ámbito donde se va a utilizar la aplicación. También recogeremos los requisitos y funcionalidades necesarias de la aplicación.
- **Evaluación prototipo baja definición:** Se evalúa con el cliente la propuesta de prototipo de baja definición realizado en la fase de prototipado.
- **Evaluación prototipo interactivo:** Se evalúa con el cliente la propuesta de prototipo interactivo realizado en la segunda iteración de la fase de prototipado.
- **Entrega final:** Se entrega el producto final al cliente.

2.3. Seguimiento

2.3.1. Evolución del proyecto

- **Hito: Inicio del proyecto:** En febrero de 2013 se asignaron los proyectos fin de carrera y se inició el proyecto en si.
- **Hito: Entrevista con el cliente:** El día 4 de marzo de 2013 se realizó la primera visita al centro Obregón, con el objetivo de obtener una visión global de las necesidades del cliente así como identificar los requerimiento del proyecto.
- **Hito: Evaluación del prototipo de baja fidelidad:** El día 6 de junio de 2013 se realizó otra visita al centro Obregón con el objetivo de evaluar el prototipo no funcional propuesto. La evaluación de este prototipo se encuentra en el capítulo 5.3.
- **Hito: Evaluación del prototipo interactivo:** Aun no realizada.
- **Hito: Entrega final:** Aun no realizada.

2.3.2. Herramientas necesitadas durante el seguimiento

Durante el desarrollo del proyecto, se ha necesitado utilizar una serie de herramientas que no estaban previstas en la planificación inicial. Son las siguientes:

- **Google Calendar:** Agenda y calendario electrónico usado en este proyecto para llevar registro de realización de hitos y de otras fechas relevantes.

2.3.3. Dificultades encontradas

En esta sección indicamos las distintas dificultades encontradas durante la realización de este proyecto. Las cuales han provocado una demora en su finalización.

Las dificultades son las siguientes:

- Inexperiencia en el uso de lenguajes de programación web.
- Dificultades en el trabajo en equipo.
- Dificultad de implementar algunos minijuegos en concreto.
- Dificultad de adaptar el prototipo web a Android.

Entorno Tecnológico

Probablemente los cambios generados por las nuevas tecnologías en la sociedad las últimas dos décadas pasen a la historia como un hito similar a la invención de la imprenta o la máquina de vapor, por su impacto en todos los ámbitos y el carácter de evolución acelerada que conllevan estos cambios.

En este capítulo se hace un repaso por algunas de las tecnologías que son usadas en este proyecto con el objetivo de saber más acerca de su funcionamiento, uso y evolución.

3.1. Android

Android es, según el sitio web de los desarrolladores de Android[17] "la plataforma móvil más popular del mundo".

Esa plataforma está compuesta por un sistema operativo, cierto middleware (servicios y soluciones que conectan los componentes de Android), y una serie de aplicaciones móviles de Google. Así, tendríamos una base que tiene como pilar el **Android Open Source Project** (AOSP), a la que se sumarían los **Google Mobile Services** (Google Play, Google Maps, Google Now ...) y las **Google Apps** (Gmail, Google Drive, Google Keep ...).

Precisamente la base de la plataforma Android es el citado AOSP, un proyecto en el que colabora la comunidad de usuarios y que ofrece una alternativa a la plataforma como tal, aunque la propia Google es la principal responsable de la gestión y desarrollo del proyecto.

AOSP podría considerarse como una plataforma abierta en sí misma, ya que además del sistema operativo móvil ofrece algunas aplicaciones nativas que permiten por ejemplo reproducir música, realizar y recibir llamadas o sacar fotos con el móvil o tablet que estamos manejando.

3.1.1. Historia de Android

Google adquiere Android Inc. en el año 2005. Se trataba de una pequeña compañía, que acababa de ser creada, orientada a la producción de aplicaciones para terminales móviles. Ese mismo año empiezan a trabajar en la creación de una máquina virtual Java optimizada para móviles, la máquina virtual Dalvik VM.

En el año 2007 se crea el consorcio Handset Alliance con el objetivo de desarrollar estándares abiertos para móviles. Está formado por Google, Intel, Texas Instruments, Motorola, T-Mobile, Samsung, Ericson, Toshiba, Vodafone, NTT DoCoMo, Sprint Nextel y otros. Una pieza clave de los objetivos de esta alianza es promover el diseño y difusión de la plataforma Android. Sus miembros se han comprometido a publicar una parte importante de su propiedad intelectual como código abierto

bajo licencia Apache v2.0. En noviembre del 2007 se lanza una primera versión del Android SDK. Al año siguiente aparece el primer móvil con Android (T-Mobile G1). En octubre Google libera el código fuente de Android principalmente bajo licencia de código abierto Apache (licencia GPL v2 para el núcleo). Ese mismo mes se abre Android Market, para la descarga de aplicaciones. Durante el año 2010 Android se consolida como uno de los sistemas operativos para móviles más utilizados, con resultados cercanos al iPhone e incluso superando al sistema de Apple en EE.UU.

En el 2011 se lanza la versión 3.x específica para tabletas y 4.x tanto para móviles como para tabletas. Durante ese año Android se consolida como la plataforma para móviles más importante, alcanzando una cuota de mercado superior al 50%. En 2012, Google cambia su estrategia en su tienda de descargas online, reemplazando el Android Market por Google Play Store, donde en un único portal, unifica la descarga de aplicaciones, contenidos e incluso dispositivos.

3.1.2. Comparativa con otras plataformas

Existen múltiples plataformas móviles en el mercado actualmente. Cada una de ellas se esfuerza al máximo para conseguir una mayor adopción, pero no todas han cosechado el mismo éxito. La web idc.com[21] nos proporciona estadísticas fiables sobre la cuota de mercado de las distintas plataformas móviles (Ver tabla 3.1).

| Android | iOS | Windows Phone | BlackBerry | Otros |
|---------|-------|---------------|------------|-------|
| 78.6% | 15.2% | 3.3% | 1.9% | 1.0% |

Tabla 3.1: Cuota de mercado de las principales plataformas móviles (año 2013)

A continuación, comparamos las tres plataformas móviles más importantes del panorama actual. La suma de estas tres plataformas supera el 95 % de la cuota de mercado total de todas las plataformas móviles existentes.

Como se puede observar en la tabla 3.2, Android tiene varias ventajas respecto a sus competidores.

Estas ventajas son las siguientes:

- **Economico:** Android es más asequible en el aspecto monetario tanto para los desarrolladores como para los usuarios.
- **Comunidad:** Al ser Android un sistema de código abierto, es más personalizable, tanto a nivel de escritorio usando alguno de los muchos widgets que existen en la tienda de aplicaciones como usando alguna de las ROMS que terceras personas han desarrollado para Android.
- **Variedad:** Existen multitud de dispositivos Android, desde los más económicos hasta los más potentes. Siempre va a existir un Android que se adapte a las necesidades de cada usuario.

3.1.3. Arquitectura de Android

La figura 3.1 muestra la arquitectura de Android. Como se puede ver está formada por cuatro capas. Una de las características más importantes es que todas las capas están basadas en software libre.

¹Según AppBrain[3]

²Segun 148apps[1]

³Según Microsoft[12]

| | Android 4.4 | iOS 7 | Windows Phone 8 |
|---|----------------------------|-------------------------|--------------------------|
| Compañía | Open Handset Alliance | Apple | Microsoft |
| Núcleo de S.O. | Linux | Mac OS X | Windows CE |
| Familia CPU soportada | ARM, MIPS, Power, X86 | ARM | ARM |
| Lenguaje de programación | Java, C++ | Objective-C, C++ | C# |
| Licencias de software | Libre y abierto | Propietaria | Propietaria |
| Año de lanzamiento | 2008 | 2007 | 2010 |
| Motor de navegador web | Webkit | Webkit | Pocket Internet Explorer |
| Soporta Flash | No, Aplicación de terceros | No | No |
| HTML 5 | Si | Si | Si |
| Tienda de aplicaciones | Play Store | App Store | Windows Phone Store |
| Numero de aplicaciones | +1.130.000 ¹ | +1.100.000 ² | +200,000 ³ |
| Coste de publicación | 25\$ Una vez | 99\$ año | 99\$ año |
| Plataforma de desarrollo | Windows, Mac, Linux | Mac | Windows |
| Interfaz personalizable | Si | No | Si |
| Actualizaciones automáticas del S.O. | Depende del fabricante | Si | Depende del fabricante |
| Soporta memoria externa | Si | No | Si |
| Fabricante único | No | Si | No |
| Variedad de dispositivos | Muy alta | Muy baja | Baja |
| Tipo de pantalla | Capacitiva | Capacitiva | Capacitiva |
| Aplicaciones nativas | Si | Si | Si |

Tabla 3.2: Comparativa de las principales plataformas móviles

El núcleo Linux

El núcleo de Android está formado por el sistema operativo Linux. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos.

Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del hardware.

Android comenzó utilizando el kernel de Linux versión 2.6, aunque mas adelante usa versiones posteriores de el núcleo Linux[40].

Las versiones del nucleo Linux usados por Android AOSP, puede consultarse en la tabla 3.3.

Runtime de Android

Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de correr Android (poca memoria y procesador limitado) no fue posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones.

Algunas características de la máquina virtual Dalvik que facilitan esta optimización de recursos son: que ejecuta ficheros Dalvik ejecutables (.dex) -formato optimizado para ahorrar memoria. Además, está basada en registros. Cada aplicación corre en su propio proceso Linux con su propia

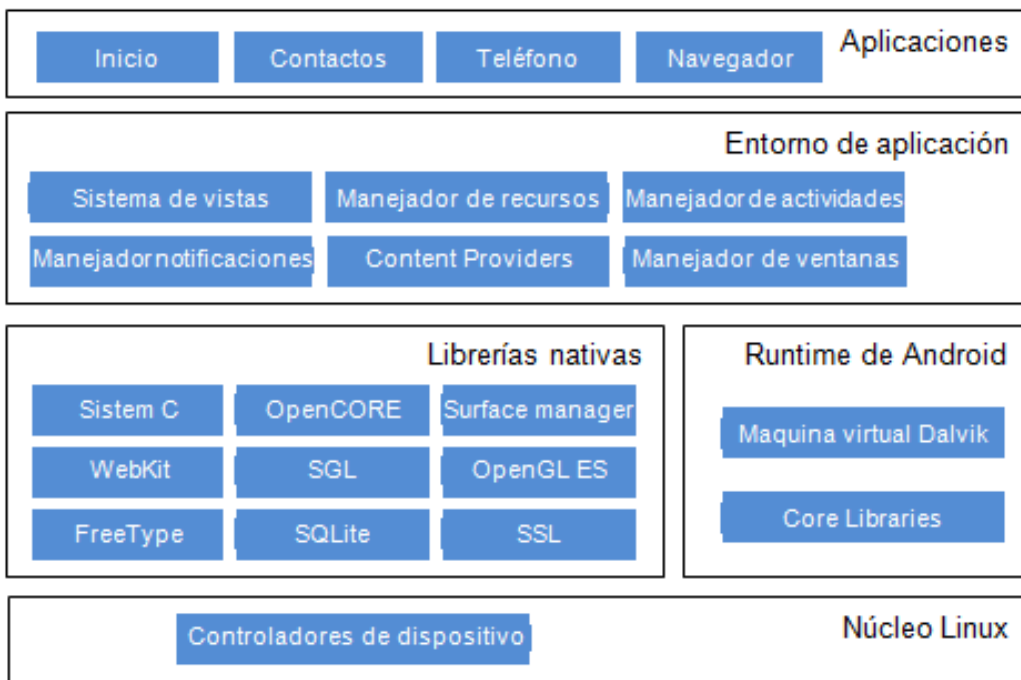


Figura 3.1: Arquitectura de Android

instancia de la máquina virtual Dalvik. Delega al kernel de Linux algunas funciones como threading y el manejo de la memoria a bajo nivel.

También se incluye en el Runtime de Android el core libraries con la mayoría de las librerías disponibles en el lenguaje Java.

La máquina virtual Dalvik

Una máquina virtual es un programa dentro de un sistema operativo que simula ser su propio sistema operativo. Para que os hagáis una idea, en Windows, que quizá nos es más familiar. Cada vez que ejecutáramos un programa este abriría un Windows propio que se ejecutara en él. Suena absurdo, pero ahora pensad que queréis ejecutar un programa para Linux en Windows o la inversa, para ello tendríais que usar una máquina virtual. En Android, el sistema operativo base es Linux y la máquina virtual es conocida como Dalvik que es quien compila los archivos Class de Java en archivos ODEX para ejecutarlos en el terminal.

La máquina virtual ART

El lanzamiento de Android 4.4 KitKat trajo entre sus muchas novedades, una nueva máquina virtual llamada ART.

ART hace referencia a Android Runtime, o tiempo de ejecución de Android, éste es el responsable de correr las aplicaciones en Android. ART es el sucesor de Dalvik Runtime y su principal diferencia es que el primero elimina la necesidad de una máquina virtual o de interpretar el código de las

| Versión Android | | Nivel API | Linux Kernel |
|-----------------|--------------------|-----------|--------------|
| 1.5 | Cupcake | 3 | 2.6.27 |
| 1.6 | Donut | 4 | 2.6.29 |
| 2.0/1 | Eclair | 5-7 | 2.6.29 |
| 2.2.x | Froyo | 8 | 2.6.32 |
| 2.3.x | Gingerbread | 9,10 | 2.6.35 |
| 3.x.x | Honeycomb | 11-13 | 2.6.36 |
| 4.0.x | Ice Cream Sandwich | 14,15 | 3.0.1 |
| 4.1.x | Jelly Bean | 16 | 3.0.31 |
| 4.2.x | Jelly Bean | 17 | 3.4.0 |
| 4.3 | Jelly Bean | 18 | 3.4.39 |
| 4.4 | KitKat | 19 | 3.4.39 |

Tabla 3.3: Versiones del kernel Linux usadas en Android AOSP

aplicaciones lo cual se ve reflejado en mejoras en el desempeño de la aplicación desde el lanzamiento. Esto lo hace por medio del compilador de código, el cual es diferente que en Dalvik.

La principal diferencia entre la antigua Dalvik y la nueva ART, reside en que la antigua ejecuta una maquina virtual interpretando el código al tiempo que se inicia la aplicación. En cambio, ART es AOT (Ahead-Of-Time), es decir, comienza una pre-compilación al instalar la aplicación y, por tanto, al ejecutarla esta ya no requiere tanta carga de datos como antes y hace que el inicio y ejecución de las aplicaciones sea mucho mas rápida.

Por supuesto, esto también afecta al tiempo de cierre de una aplicación. Esta será consecuentemente más rápida, y al mismo tiempo liberará el sistema del uso intenso durante largos periodos, lo cual, significa que el procesador trabajará menos y por tanto se ahorra batería a largo plazo.

A pesar de todas estas ventajas, ART, también tiene inconvenientes. Primero, que al incorporar una parte de la compilación en la instalación de una app, el tiempo para instalarla se verá aumentado. Y segundo, que ya que se añade cierto código durante la instalación, las aplicaciones ocuparán algo más, sin embargo, cuantas más aplicaciones instaladas, mayor será esta carga de espacio.

Librerías nativas

Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son:

- **System C library:** una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.
- **Media Framework:** librería basada en PacketVideo's OpenCORE; soporta codecs de reproducción y grabación de multitud de formatos de audio vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- **Surface Manager:** maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- **WebKit:** soporta un moderno navegador web utilizado en el navegador Android y en la vista webview. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.

- **SGL:** motor de gráficos 2D.
- **Librerías 3D:** implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- **FreeType:** fuentes en bitmap y renderizado vectorial.
- **SQLite:** potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- **SSL:** proporciona servicios de encriptación Secure Socket Layer.

Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones (sensores, localización, servicios, barra de notificaciones,). También se conoce como Java SDK. La arquitectura ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas (sujetas a las restricciones de seguridad). Este mismo mecanismo permite a los usuarios reemplazar componentes. Una de las mayores fortalezas del entorno de aplicación de Android es que se aprovecha el lenguaje de programación Java. El SDK de Android no acaba de ofrecer todo lo disponible para su estándar del entorno de ejecución.

- **Views:** extenso conjunto de vistas, (parte visual de los componentes).
- **Resource Manager:** proporciona acceso a recursos que no son en código.
- **Activity Manager:** maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- **Notification Manager:** permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- **Content Providers:** mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos).

Aplicaciones

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de correr en la máquina virtual Dalvik para garantizar la seguridad del sistema.

Normalmente las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java podemos utilizar el Android SDK. Existe otra opción consistente en desarrollar las aplicaciones utilizando C/C++. Para esta opción podemos utilizar el Android NDK (Native Development Kit).

3.1.4. Versiones de Android y niveles de API

Según explica Jesús Tomás Girones en su libro "El gran libro de Android"[23], hay que elegir la versión de Android para la que queremos desarrollar. Ya que hay clases y métodos en Android disponibles a partir de una versión, por consiguiente, si las vamos a usar, hay que conocer la versión mínima necesaria.

Cuando se ha lanzado una nueva plataforma, siempre ha sido compatible con las versiones anteriores. Es decir, solo se añaden nuevas funcionalidades y, en el caso de modificar alguna funcionalidad, no se elimina, se etiqueta como obsoletas pero se pueden continuar utilizando.

A continuación se describen las plataformas lanzadas hasta la fecha con una breve descripción de las novedades introducidas. Las plataformas se identifican de tres formas alternativas: versión, nivel de API y nombre. EL nivel de API corresponde a números enteros comenzando desde el 1. Para los nombres se han elegido postres en orden alfabético. Las dos primeras versiones, que corresponden a las letras A y B no recibieron nombre.

Android 1.0 Nivel de API 1

Primera versión de Android. Publicada en septiembre de 2008. Nunca se utilizó comercialmente, por lo que no tiene sentido desarrollar para esta plataforma.

Android 1.1 Nivel de API 2

Publicada en febrero de 2009. No se añadieron apenas funcionalidades, simplemente se corrigieron algunos errores de la versión anterior. No existen usuarios utilizando esta versión.

Android 1.5 Nivel de API 3 (Cupcake)

Publicada en abril de 2009. Como novedades, se incorporaron la posibilidad de teclado en pantalla con predicción de texto, los terminales ya no tienen que tener un teclado físico, así como la capacidad de grabación avanzada de audio y vídeo. También aparecen los widgets de escritorio y live folders. Incorpora soporte para bluetooth estéreo, por lo que permite conectarse automáticamente a auriculares bluetooth. Las transiciones entre ventanas se realizan mediante animaciones.

Android 1.6 Nivel de API 4 (Donut)

Publicada en diciembre de 2009. Permite capacidades de búsqueda avanzada en todo el dispositivo. También se incorpora *gestures* y *multi-touch*. Permite la síntesis de texto a voz. También facilita que una aplicación pueda trabajar con diferentes densidades de pantalla. Soporte para resolución de pantallas VVVGGA. Aparece un nuevo atributo XML, *onClick*, que puede especificarse en una vista. Se mejora Play Store, antes Android Market permitiendo una búsqueda más sencilla de aplicaciones. Soporte para CDMA/EVDO, 802.1x y VPNs. Mejoras en la aplicación de la cámara.

Android 2.0 Nivel de API 5 (Éclair)

Publicada en octubre de 2009. Esta versión de API apenas ha sido utilizada por los usuarios, dado que la mayoría de fabricantes pasaron directamente de la versión 1.6 a la 2.1. Como novedades cabría destacar que incorpora un API para manejar el bluetooth 2.1. Nueva funcionalidad que permite sincronizar adaptadores para conectarlo a cualquier dispositivo. Ofrece un servicio centralizado de manejo de cuentas. Mejora la gestión de contactos y ofrece más ajustes en la cámara. Se ha optimizado la velocidad de hardware. Se aumenta el número de tamaños de ventana y resoluciones soportadas. Nueva interfaz del navegador y soporte para HTML5. Mejoras en el calendario y soporte para Microsoft Exchange. La clase *MotionEvent* ahora soporta eventos en pantallas multitáctil.

Android 2.1 Nivel de API 7

Publicada en enero de 2010. Se considera una actualización menor, por lo que le siguieron llamando Éclair. Destacamos el reconocimiento de voz que permite introducir un campo de texto dictando sin necesidad de utilizar el teclado. También permite desarrollar fondos de pantalla animados. Se puede obtener información sobre la señal de la red actual que posea el dispositivo. En el paquete WebKit se incluyen nuevos métodos para manipular bases de datos almacenadas en Web. También se permite obtener permisos de geolocalización, y modificarlos en WebView. Se incorporan mecanismos para administrar la configuración de la caché de aplicaciones, almacenamiento web, y modificar la resolución de la pantalla. También se puede manejar vídeo, historial de navegación, vistas personalizadas, etc.

Android 2.2 Nivel de API 8 (Froyo)

Publicada en mayo de 2010. Es la primera versión que aun cuenta con algun usuario (Un 1.3% en enero de 2014). Como característica más destacada se puede indicar la mejora de velocidad de ejecución de las aplicaciones (ejecución del código de la CPU de 2 a 5 veces más rápido que en la versión 2.1 de acuerdo a varios *benchmarks*). Esto se consigue con la introducción de un nuevo compilador JIT de la máquina Dalvik.

Se añaden varias mejoras relacionadas con el navegador Web, como el soporte de Adobe Flash 10.1 y la incorporación del motor Javascript V8 utilizado en Chrome o la incorporación del campo de "subir fichero" en un formulario.

El desarrollo de aplicaciones permite las siguientes novedades: se puede preguntar al usuario si desea instalar una aplicación en un medio de almacenamiento externo (como una tarjeta SD), como alternativa a la instalación en la memoria interna del dispositivo. Las aplicaciones se actualizan de forma automática cuando aparece una nueva versión. Proporciona un servicio para la copia de seguridad de datos que se puede realizar desde la propia aplicación para garantizar al usuario el mantenimiento de sus datos. Por último, se facilita que las aplicaciones interaccionen con el reconocimiento de voz y que terceras partes proporcionen nuevos motores de reconocimiento.

Se mejora la conectividad: ahora podemos utilizar nuestro teléfono para dar acceso a Internet a otros dispositivos (*tethering*), tanto por USB como por Wi-Fi. También se añade el soporte a Wi-Fi IEEE 802.11n.

Se añaden varias mejoras en diferentes componentes: En el API gráfica OpenGL ES se pasa a soportar la versión 2.0. También se puede realiza; fotos o vídeos en cualquier orientación (incluso vertical) y configurar otros ajustes de la cámara. Para finalizar, permite definir modos de interfaz de usuario ("automóvil" y "noche") para que las aplicaciones se configuren según el modo seleccionado por el usuario.

Android 2.3 Nivel de API 9 (Gingerbread)

Publicada en diciembre de 2010. Debido al éxito de Android en las nuevas tabletas ahora soporta mayores tamaños de pantalla y resoluciones (WXGA y superiores).

Incorpora un nuevo interfaz de usuario con un diseño actualizado. Dentro de las mejoras de la interfaz de usuario destaca la mejora de la funcionalidad de "cortar, copiar y pegar" y un teclado en pantalla con capacidad multitáctil.

Se incluye soporte nativo para varias cámaras, pensado en la segunda cámara usada en videoconferencia. La incorporación de esta segunda cámara ha propiciado la inclusión de reconocimiento facial para identificar el usuario del terminal.

La máquina virtual de Dalvik para Android introduce un nuevo recolector de basura que minimiza las pausas de la aplicación, ayudando a garantizar una mejor animación y el aumento de la capacidad de respuesta en juegos y aplicaciones similares. Se trata de corregir así una de las lacras de este sistema operativo móvil, que en versiones previas no ha sido capaz de cerrar bien las aplicaciones en desuso. Se dispone de mayor apoyo para el desarrollo de código nativo (NDK). También se mejora la gestión de energía y control de aplicaciones. Y se cambia el sistema de ficheros, que pasa de YAFFS a EXT4.

Entre otras novedades destacamos en soporte nativo para telefonía sobre Internet VoIP/SIP. El soporte para reproducción de vídeo WebM/VP8 y codificación de audio AAC. El soporte para la tecnología NFC. Las facilidades en el audio, gráficos y entradas para los desarrolladores de juegos. El soporte nativo para más sensores (como giroscopios y barómetros). Un gestor de descargas para las descargas largas.

Android 3.0 Nivel de API 11 (Honeycomb)

Publicada en febrero de 2011. Para mejorar la experiencia de Android en las nuevas tabletas se lanza la versión 3.0 optimizada para dispositivos con pantallas grandes. La nueva interfaz de usuario ha sido completamente rediseñada con paradigmas nuevos para la interacción, navegación y personalización. La nueva interfaz se pone a disposición de todas las aplicaciones, incluso las construidas para versiones anteriores de la plataforma.

Las principales novedades de este SDK son: Con el objetivo de adaptar la interfaz de usuario a pantallas más grandes se incorporan las siguientes características: resolución por defecto WXGA (1280x800), escritorio 3D con widgets rediseñados, nuevos componentes y vistas, notificaciones mejoradas, arrastrar y soltar, nuevo cortar y pegar, barra de acciones para que las aplicaciones dispongan de un menú contextual siempre presente y otras características para aprovechar las pantallas más grandes.

Se mejora la reproducción de animaciones 2D/3D gracias al renderizador OpenGL acelerado por hardware. El nuevo motor de gráficos Rederscript saca un gran rendimiento de los gráficos en Android e incorpora su propia API.

Primera versión de la plataforma que soporta procesadores multinúcleo. La máquina virtual Dalvik ha sido optimizada para permitir multiprocesado, lo que permite una ejecución más rápida de las aplicaciones, incluso aquellas que son de hilo único.

Se incorporan varias mejoras multimedia, como listas de reproducción M3U a través de HTTP Live streaming, soporte a la protección de derechos musicales (DRM) y soporte para la transferencia de archivos multimedia a través de USB con los protocolos MTP y PTP.

En esta versión se añaden nuevas alternativas de conectividad, como las nuevas APIS de Bluetooth A2DP y HSP con streaming de audio. También, se permite conectar teclados completos por USB o Bluetooth.

El uso de los dispositivos en un entorno empresarial es mejorado. Entre las novedades introducidas destacamos las nuevas políticas administrativas con encriptación del almacenamiento, caducidad de contraseña y mejoras para administrar los dispositivos de empresa de forma eficaz.

A pesar de la nueva interfaz gráfica optimizada para tabletas, Android 3.0 es compatible con las aplicaciones creadas para versiones anteriores. La tecla de menú, inexistente en las nuevas tabletas, es reemplazada por un menú que aparece en la barra de acción.

Android 3.1 Nivel de API 12

Publicada en mayo de 2011. Permite manejar dispositivos conectados por USB (tanto host como dispositivo) Protocolo de transferencia de fotos y vídeos (PTP/MTP) y de tiempo real /RTP)

Android 3.2 Nivel de API 13

Publicada en julio de 2011. Optimizaciones para distintos tipos de tableta. Zoom compatible para aplicaciones de tamaño fijo. Sincronización multimedia desde SD

Android 3.0 Nivel de API 14 (Ice Cream Sandwich)

Publicada en octubre de 2011. La característica mas importante es que se unifican las dos versiones anteriores (2.x para teléfonos y 3.x para tabletas) en una sola compatible con cualquier tipo de dispositivo. Entre las características mas importantes destacamos:

Se introduce un nuevo interfaz de usuario totalmente renovado. Nuevo API de reconocedor facial que permite, entre otras muchas aplicaciones, desbloquear el teléfono. También se mejora en el reconocimiento de voz.

Aparece un nuevo gestor de trafico de datos por Internet, donde podemos ver el consumo en una gráfica y donde podemos definir los limites a ese consumo para evitar cargos inesperados con la operadora. Incorpora herramientas para la edición de imágenes en tiempo real, con herramientas para distorsionar, manipular e interactuar con la imagen en el momento de ser capturada. Se mejora la API para comunicaciones NFC y la integración con redes sociales.

En diciembre de 2011 aparece una actualización de mantenimiento (versión 4.0.2) que no aumenta el nivel de API.

Android 4.0.3 Nivel de API 15

Publicada en diciembre de 2011. Se introducen ligeras mejoras en algunas APIs incluyendo el de redes sociales, calendario, revisor ortográfico, texto a voz y bases de datos entre otros. En marzo de 2012 aparece la actualización 4.0.4.

Android 4.1 Nivel de API 16 (Jelly Bean)

Publicada en julio de 2012. En esta versión se hace hincapié en mejorar un punto débil de Android: la fluidez del interfaz de usuario. Con este propósito se incorporan varias técnicas, como: sincronismo vertical, triple búfer y aumentar la velocidad del procesador al tocar la pantalla.

Se mejoran las notificaciones con un sistema de información expandible personalizada. Los Widgets de escritorio pueden ajustar su tamaño y hacerse sitio de forma automática al situarlos en el escritorio. El dictado por voz puede realizarse sin conexión a Internet (de momento en ingles).

Se introducen varias mejoras en Google Search. Se potencia la búsqueda por voz con resultados en forma de ficha. La función Google Now permite utilizar información de posición, agenda y hora en las búsquedas.

Se incorporan nuevo soporte para usuarios internacionales: como texto bidireccional y teclados instalables. Para mejorar la seguridad las aplicaciones son cifradas. También se permite actualizaciones parciales de aplicaciones.

Android 4.2 Nivel de API 17

Publicada en noviembre de 2012. Una de las novedades más importantes es que podemos crear varias cuentas de usuario en el mismo dispositivo. Aunque, esta característica solo está disponible en tabletas. Cada cuenta tendrá sus propias aplicaciones y configuración.

Los Widgets de escritorio pueden aparecer en la pantalla de bloqueo. Se incorpora un nuevo teclado predictivo deslizante al estilo Swype. Posibilidad de conectar dispositivo y TVHD mediante wifi (Miracast). Mejoras menores en las notificaciones. Nueva aplicación de cámara que incorpora la funcionalidad Photo Sphere para hacer fotos panorámicas inmersivas (en 360°).

Android 4.3 Nivel de API 18

Publicada en julio de 2013. Han mejorado la aplicación Teléfono de Android para añadir la opción de autocompletar con el teclado de marcación. Esta nueva opción nos permite ver sugerencias de números y nombres que vayan coincidiendo con el número que estamos marcando. La cámara tiene nueva interfaz, en el que muestra las opciones en forma de arco en vez de círculo para que no se tapen con el dedo, permite hacer fotos con los botones de volumen y añaden la opción de temporizador.

Da soporte oficial a OpenGL ES 3.0, a la API más reciente de esta potente librería gráfica. Los dispositivos que tengan en su hardware una GPU compatible disfrutarán de gráficos más realistas, con texturas y efectos de la actual generación de consolas, como mejores sombras a tiempo real, mejor iluminación, partículas, destellos o reflejos.

Se da soporte al Bluetooth Smart en dispositivos compatibles (Bluetooth 4.0). También conocido como Bluetooth LE (de bajo consumo) permite las aplicaciones detecten y se comuniquen fácilmente con dispositivos con Bluetooth Smart y consumir poca batería.

También dan soporte Bluetooth AVRCP 1.3 para poder dar más información a los dispositivos conectados.

Android 4.4 Nivel de API 19 (KitKat)

Publicada en octubre de 2013. Esta versión de Android está diseñada para funcionar todavía más rápido y suave en una gama más amplia de dispositivos. Han optimizado Android 4.4 para que funcione perfectamente en dispositivos con 512 MB de RAM. Añade un administrador de impresión nativo que media entre las aplicaciones y los servicios de impresión. Permite que las aplicaciones puedan acceder fácilmente a los archivos que tenemos guardado en la nube de cualquier servicio, para lograr una mejor integración. Podremos acceder nuestras carpetas y archivos de Google Drive, DropBox, Box, etc con la misma facilidad que en las carpetas y archivos que tenemos en la memoria.

Se añade soporte aquellos dispositivos que en su hardware soporte dosificación de los sensores, algo que hará que se gaste mucho menos batería, ya que recoge y entrega eventos del sensor de manera eficiente en lotes, en lugar de individualmente a medida que se detectan, controlando la frecuencia.

Añade también el modo a pantalla completa para cualquier aplicación. Este modo ocultará tanto la barra de estado como la barra de navegación en aquellos dispositivos que no tengan botones físicos para que la aplicación ocupe toda la pantalla. Con un simple gesto, un toque en los bordes superior o inferior, aparecerán los botones.

En el apartado de accesibilidad Android 4.4 ha implementado el apartado de subtítulos, para que el usuario diga en que idioma, tamaño y estilo quiere ver los subtítulos de los vídeos. Todas las aplicaciones que muestran subtítulos en sus vídeos tendrían que respetar esa configuración.

3.1.5. Elección de la versión de API

A la hora de seleccionar la plataforma de desarrollo hay que consultar si necesitamos alguna característica especial que solo esté disponible a partir de una versión. Todos los usuarios con versiones inferiores a la seleccionada no podrán instalar la aplicación. Por lo tanto, es recomendable seleccionar la menor versión posible que nuestra aplicación pueda soportar.

Por lo tanto, para dejar fuera de nuestra aplicación al menor número de usuarios posibles, debemos saber cuántos usuarios tiene cada versión de API. Para determinar ese número de usuarios, Google nos brinda una web donde consultar este dato. En la página de Android developers[18] podemos obtener la estadística (Figura 3.2) de dispositivos Android según plataforma instalada, que han accedido a Google Play Store en los últimos 7 días.

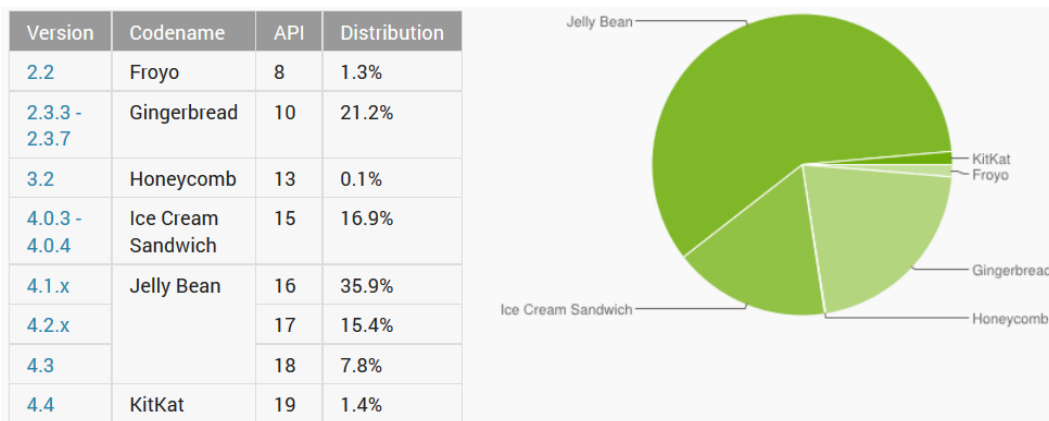


Figura 3.2: Dispositivos Android según plataforma instalada, a febrero de 2014

3.2. HTML 5

Lenguaje de marcado que proporciona la estructura donde se irán colocando los elementos de nuestra aplicación. De HTML 5 se utilizará profusamente el elemento **Canvas**, ya que este elemento proporciona un lienzo donde se define en JavaScript cada uno de los trazos a dibujar, mediante coordenadas.

3.2.1. La historia de HTML5

HTML4, que se dio por cerrado en 1998, es el lenguaje de marcado que conforma la base de la gran mayoría de las páginas web que podemos ver a día de hoy. Los diseñadores y desarrolladores web han estado utilizando la especificación HTML 4.01 durante bastantes años de manera satisfactoria, combinándola con CSS para la definición de estilos y con JavaScript para añadir interactividad a los contenidos.

Tras la finalización de HTML 4.0.1, el W3C continuó sus trabajos en consonancia con la evolución de la web, y comenzó con un lenguaje llamado XHTML 1.0. Existe una pequeña diferencia entre HTML 4.0.1 y XHTML 1.0 (XHTML es un lenguaje más estricto: por ejemplo, todas las etiquetas deben indicarse en minúsculas). Uno de los objetivos de XHTML 1.0 era crear un lenguaje

de markup que pudiera extenderse y resolver las necesidades de las tecnologías futuras, por ejemplo para los dispositivos móviles. Muchos sitios web se han creado utilizando XHTML 1.0 como marco de desarrollo y muchos diseñadores y desarrolladores valoran positivamente sus reglas sintácticas más estrictas.

En paralelo a la adopción y difusión de XHTML 1.0, se empezó con el desarrollo de otro lenguaje, XHTML 2.0, que añadía una serie de novedades de gran interés a las páginas web y que se apoyaba de forma notoria en XML. No obstante, este nuevo lenguaje adolecía de diversos problemas de tipo técnico y no resolvía adecuadamente las necesidades reales de los desarrolladores.

En 2004, un grupo de representantes de los principales fabricantes de navegadores y un grupo de trabajo de desarrolladores web formaron un grupo independiente llamado WHATWG (iniciales de Web Hypertext Application Technology Working Group). Su misión consistía en crear una especificación de lenguaje HTML mejor, orientada a crear un nuevo tipo de aplicaciones web pero manteniendo la compatibilidad con las versiones en activo de los navegadores existentes.

Durante unos dos años y medio el W3C y el grupo WHATWG trabajaron en paralelo y de manera independiente hasta que en 2006, Tim Berners-Lee, creador de la World Wide Web y fundador del W3C anunció que el W3C y WHATWG trabajarían juntos en la elaboración del estándar.

Como resultado de todo ello, se abandonó el desarrollo de XHTML 2.0 y la especificación HTML 4.0 se reformó con el nombre HTML5. No obstante, conviene recordar que HTML5 empezó siendo Web Applications 1.0, y eso nos da una pista muy clara para entender para qué fue diseñado realmente HTML5.

Gracias a que el W3C tomó el liderazgo para el desarrollo de HTML5, ahora los desarrolladores tenemos la garantía de que HTML5 será realmente un estándar gratuito y con las suficientes garantías de no incurrir en problemas de propiedad intelectual.

3.2.2. El elemento Canvas

El elemento canvas puede definirse como un entorno para crear imágenes dinámicas.

Permite especificar un área de la página donde se puede, a través de scripts, dibujar y renderizar imágenes, lo que amplía notablemente las posibilidades de las páginas dinámicas y permite hacer cosas que hasta ahora estaban reservadas a los desarrolladores en Flash, con la ventaja que para usar canvas no será necesario ningún plugin en el navegador, lo que mejorará la disponibilidad de esta nueva aplicación.

Canvas permite dibujar en la página y actualizar dinámicamente estos dibujos, por medio de scripts y atendiendo a las acciones del usuario. Todo esto da unas posibilidades de uso tan grandes como las que disponemos con el plugin de Flash, en lo que respecta a renderización de contenidos dinámicos. Las aplicaciones pueden ser grandes como podamos imaginar, desde juegos, efectos dinámicos en interfaces de usuario, editores de código, editores gráficos, aplicaciones, efectos 3D, etc.

Inicialmente lo implementó Apple para Safari. Luego fue adoptado por otros navegadores como Mozilla Firefox a partir de la versión 1.5. Por lo que respecta a chrome, es un navegador que utiliza el mismo motor renderizado que Safari, por lo que también soporta el elemento Canvas. Por otra parte, Internet Explorer no comenzó a dar soporte a canvas hasta la versión 9.

En la tabla 3.4 puede observarse el soporte de los distintos navegadores al componente canvas según los datos que ofrece la pagina caniuse.com[19].

| Característica | Canvas (Soporte básico) | WebGL 3D Canvas graphics | Canvas blend modes | Text API for Canvas |
|--------------------|-------------------------|--------------------------|--------------------|---------------------|
| IE | Si | Si | No | Si |
| Firefox | Si | Parcial | Si | Si |
| Chrome | Si | Si | Si | Si |
| Safari | Si | Parcial | Si | Si |
| iOS Safari | Si | No | Si | Si |
| Opera mini | Parcial | No | No | No |
| Android Browser | Si | No | Si | Si |
| Blackberry Browser | Si | Si | No | Si |
| IE Mobile | Si | No | No | Si |

Tabla 3.4: Soporte del elemento canvas segun distintos navegadores

3.3. Formatos de imagen

Las imágenes digitales se pueden guardar en distintos formatos.[36] Cada uno se corresponde con una extensión específica del archivo que lo contiene. Los más utilizados en la actualidad son: GIF, JPG, PNG y el prometedor WebP.

3.3.1. GIF

El formato GIF (Graphics Interchange Format - Formato de Intercambio Gráfico) puede contener hasta 256 colores. Por esta razón, GIF no es un buen formato para guardar fotografías que puedan tener más que esa cantidad de colores, puesto que perderían gravemente su calidad.

Es útil para guardar imágenes que tengan grandes bloques de colores planos como ilustraciones simples o logotipos para alguna compañía. Además, permite preservar la transparencia.

- Ha sido diseñado específicamente para comprimir imágenes digitales.
- Reduce la paleta de colores a 256 colores como máximo (profundidad de color de 8 bits).
- Admite gamas de menor número de colores y esto permite optimizar el tamaño del archivo que contiene la imagen.
- Admite animaciones.
- Ventaja: Es un formato idóneo para publicar dibujos en la web.
- Inconveniente: No es recomendable para fotografías de cierta calidad ni originales ya que el color real o verdadero utiliza una paleta de más de 256 colores.

3.3.2. JPEG

JPEG (Joint Photographic Experts Group - Grupo de Expertos Fotográficos Unidos) es el formato gráfico más usado no sólo en Internet sino también fuera de él. La mayor ventaja que tiene es su capacidad de compresión, que puede llegar a valores tan altos como 100:1 (una imagen de 100KB almacenada en un archivo de 1KB).

Ademas, el formato tiene las siguientes características:

- A diferencia del formato GIF, admite una paleta de hasta 16 millones de colores.
- Es el formato más común junto con el GIF para publicar imágenes en la web.
- La compresión JPEG puede suponer cierta pérdida de calidad en la imagen. En la mayoría de los casos esta pérdida se puede asumir porque permite reducir el tamaño del archivo y su visualización es aceptable. Es recomendable utilizar una calidad del 60-90% del original.
- Cada vez que se modifica y guarda un archivo JPEG, se puede perder algo de su calidad si se define cierto factor de compresión.
- Las cámaras digitales suelen almacenar directamente las imágenes en formato JPEG con máxima calidad y sin compresión.
- Ventaja: Es ideal para publicar fotografías en la web siempre y cuando se configuren adecuadamente dimensiones y compresión.
- Inconveniente: Si se define un factor de compresión se pierde calidad. Por este motivo no es recomendable para archivar originales.

3.3.3. PNG

PNG (Portable Network Graphic = Gráfico portable para la red), es un formato sin pérdida y que admite transparencia. El problema es que no alcanza una compresión tan alta como JPEG, así que sólo se usa con imágenes pequeñas y con pocos colores.

- Tiene una tasa de compresión superior al formato GIF (+10%)
- Admite la posibilidad de emplear un número de colores superior a los 256 que impone el GIF.

3.3.4. WebP

WebP (Web-Picture) Es un nuevo codec de imagen desarrollado por Google para ser usado principalmente en la web ya que auna las ventajas de los formatos JPG, PNG y GIF.

Este formato pretende ser referente respecto a la publicación de imágenes para la web. Por ello reúne en un único formato las mejores características de sus rivales. Estas características son:

- Compresión con pérdidas para reducir el tamaño de archivo,
- Compresión sin pérdidas para mantener la calidad
- Soporta transparencias
- Soporta animaciones

Frente a PNG, WebP lo mejora por tener un **mayor abanico de filtros sin pérdida**. El filtro de predicción tiene más modos (cada modo valora de forma distinta los píxeles adyacentes al que queremos calcular su valor teórico), lo que permite un mejor ajuste y por lo tanto se pueden comprimir mejor los datos. Además, hay otros dos filtros adicionales: transformación de color e indexación de colores.

Estos filtros se pueden aplicar al mismo tiempo sobre una imagen, de forma que el formato WebP ofrece una compresión máxima mayor usando técnicas que PNG no permite.

Por otro lado, frente a JPEG, WebP destaca por la posibilidad de añadir los filtros sin pérdida que acabo de comentar. Es cierto que JPEG tiene métodos de compresión sin pérdida, pero no son tan efectivos como los de WebP (y son menos usados).

Otra ventaja de WebP es que incluye métodos de compresión mejorados. Al igual que PNG, usa DEFLATE, pero además añade una caché de los colores más usados. En este sentido es el más avanzado de los cuatro formatos.

El formato WebP soporta los siguientes navegadores:

- Chrome
- Opera
- Android Browser
- Opera mobile

Aunque este formato solo sea admitido por una reducida selección de navegadores, su revolucionario formato haya que tenerlo en cuenta para un futuro.

3.3.5. Conclusiones

En la tabla 3.5 puede observarse el soporte de los distintos navegadores a los distintos tipos de codificación de imágenes.

| Codificación | GIF | JPEG | PNG | WebP |
|--------------------|-----|------|-----|------|
| IE | Si | Si | Si | No |
| Firefox | Si | Si | Si | No |
| Chrome | Si | Si | Si | Si |
| Safari | Si | Si | Si | No |
| Opera | Si | Si | Si | Si |
| iOS Safari | Si | Si | Si | No |
| Opera mobile | Si | Si | Si | No |
| Android Browser | Si | Si | Si | Si |
| Blackberry Browser | Si | Si | Si | No |
| IE Mobile | Si | Si | Si | No |

Tabla 3.5: Soporte de codecs de imagenes segun distintos navegadores

3.4. Formatos de audio y vídeo en HTML5

Las etiquetas `<audio>` y `<video>` proporcionan soporte para la reproducción de audio y video sin necesidad de plug-ins. Se utilizan codecs de video y audio para manejar vídeo y audio, y los diferentes codecs ofrecen diferentes niveles de compresión y calidad. Un formato contenedor se utiliza para almacenar y transmitir el vídeo y el audio codificado a la vez. Existen muchos codecs y formatos

contenedor, e incluso hay mas combinaciones de ellos. Sin embargo, para su uso en la web, sólo un puñado de combinaciones son relevantes.

No todos los navegadores soportan los mismos formatos para audio y video en HTML5, principalmente debido a problemas de patentes. El área de formatos de medios en la web ha sufrido mucho por la ley de patentes en muchos países, incluyendo EE.UU y la UE.

Para hacer un vídeo de HTML5, que trabaja en las nuevas versiones de los principales navegadores, puede usar el elemento `<source>` para reproducir en formatos WebM o el formato AAC MPEG H.264.

A continuación se comentan algunos de los formatos de vídeo y audio mas utilizados

3.4.1. AVI

AVI (Audio Video Interleaved = Audio y Video Intercalado) es un formato...

Ademas tiene las siguientes características:

- Es el formato estándar para almacenar vídeo digital.
- Cuando se captura vídeo desde una cámara digital al ordenador, se suele almacenar en este formato con el códec DV (Digital Video).
- El archivo AVI puede contener vídeo con una calidad excelente. Sin embargo el peso del archivo resulta siempre muy elevado.
- Admite distintos códec de compresión como CinePak, Intel Indeo 5, DV, etc. Los códec con más capacidad de compresión y una calidad aceptable son DivX y XviD.
- El formato AVI puede ser visualizado con la mayoría de reproductores: Windows Media, QuickTime, etc. siempre y cuando se encuentren instalados en el equipo los adecuados códec para cada tipo de reproductor.
- Es ideal para guardar vídeos originales que han sido capturados de la cámara digital (codificados con DV).
- No es recomendable publicarlos en Internet en este formato por su enorme peso.

3.4.2. MPEG

Es un formato estándar para la compresión de video digital. Son archivos de extensión *.MPG ó *.MPEG. Admite distintos tipos de códec de compresión: MPEG-1 (calidad CD), MPEG-2 (calidad DVD), MPEG-3 (orientado al audio MP3) y MPEG-4 (más orientado a la web). Se reproducen con Windows Media Player y QuickTime.

3.4.3. MP4 H.264 (AAC o MP3)

El formato contenedor MP4 con el códec de vídeo H.264 y, o bien el códec de audio AAC o el códec de audio MP3 es nativamente compatible con Internet Explorer, Firefox, Safari y Chrome, pero Chromium y Opera no son compatibles con el formato.

Los formatos MPEG están cubiertos por patentes, que no tienen licencia libre. Todas las licencias necesarias se pueden comprar de MPEG LA. Desde H.264 no está en un formato libre de regalías, no es apto para la plataforma web de código abierto, de acuerdo con Mozilla, Google y Opera. Sin

embargo, los formatos libres no son compatibles con Internet Explorer y Safari, Mozilla ha decidido apoyar el formato de todos modos, y Google no cumplió su promesa de eliminar el soporte para él en Chrome.

3.4.4. WebM

El formato WebM se basa en una versión restringida del formato contenedor Matroska. Siempre utiliza el códec de vídeo VP8 y el códec de audio Vorbis. WebM es soportado nativamente en Gecko (Firefox), Chrome y Opera, y el soporte para el formato se puede agregar a Internet Explorer y Safari mediante la instalación de un add-on.

El formato WebM, específicamente el códec de vídeo VP8, habría sido acusado de infracción de derechos de patente por un grupo de empresas de responder a una llamada por la MPEG LA para la formación de un consorcio de patentes, pero MPEG LA se ha comprometido a licenciar las patentes de Google bajo "transferible y de licencia gratuita". Esto significa, efectivamente, que todas las patentes conocidas en el formato WebM tienen licencia para todo el mundo de forma gratuita.

3.5. ¿Qué es Canvas?

Canvas es un elemento de HTML 5 que permite dibujar gráficos, manipular imágenes y realizar animaciones en una página web de forma dinámica. Es decir, el gráfico se crea en el momento en que se carga la página. El gráfico se define mediante programación, usualmente Javascript. Una vez creado el gráfico se pueden programar acciones para que el usuario interactúe con él, por ejemplo, haciendo clic para arrancar una animación o arrastrando un elemento del gráfico.

Aunque el elemento Canvas se crea con la etiqueta `<canvas></canvas>` al hablar de Canvas en sentido amplio nos referimos a toda la API que incluye un conjunto de funciones para dibujar, líneas, rectángulos, círculos, etc. así como para rotar, escalar, transformar elementos gráficos, y otras funciones de variado uso.

Al definir un Canvas en una página web se crea un lienzo de dibujo o área de dibujo rectangular. Después de crear el dibujo lo que queda es un "mapa de bits", es decir, cada coordenada del lienzo tiene asignado un color, no queda ninguna estructura de lo que contiene el lienzo. Canvas no crea objetos vectoriales al estilo de otros entornos como SVG o Flash, sino mapas de bits como una imagen fotográfica.

De hecho podemos cargar una imagen fotográfica en el Canvas y manipular sus píxeles uno a uno.

Canvas ha sido incorporado recientemente al HTML y no puede ser representado por los navegadores antiguos. Se requiere una versión actualizada de Chrome (26), Firefox (20), Internet Explorer (10), Safari (6.0), etc. Es conveniente dar un contenido alternativo para los navegadores que no soportan Canvas, como una imagen fija o un texto aclaratorio.

3.6. Javascript

El lenguaje de programación JavaScript, según el libro "JavaScript Guía Práctica"[16], se utiliza en las páginas web para incrementar la funcionalidad de las mismas y la interacción con el usuario, lo que se traduce en unas páginas más dinámicas. Algunas de sus funcionalidades pueden ser entre otras:

- Validar los datos introducidos en un formulario
- Detectar el navegador que está utilizando el usuario
- Almacenar información del usuario para que no tenga que volver a ser introducida (Cookies).

A principios de los 90 se empezaron a necesitar una serie de funcionalidades en las páginas que el HTML a secas no era capaz de satisfacer. Un caso muy claro fue la validación de formularios en el servidor web, el cual devolvía una respuesta al usuario incluyendo los errores encontrados en los datos, si los hubiera. Dado que por entonces la velocidad de conexión a Internet no era muy alta, surgió la necesidad de comprobar los campos de los formularios desde el propio navegador del usuario. Con esto se conseguían fundamentalmente dos cosas:

- Evitar largas esperas al usuario con la respuesta del servidor.
- Liberar al servidor de realizar este tipo de operaciones con lo que la respuesta era más rápida.

Brendan Eich, durante su trabajo en la empresa Netscape Communications, desarrolló un lenguaje capaz de dar soluciones a este problema, apareciendo por primera vez en el navegador Netscape 2.0 con gran éxito. Inicialmente lo llamaron Mocha, más tarde LiveScript y finalmente fue rebautizado como JavaScript cuando Netscape se alió con Sun Microsystems (creadores del lenguaje Java).

El código JavaScript es interpretado directamente por el navegador que se está utilizando. Esto hace que sea un lenguaje del lado del cliente, puesto que el servidor web no interpreta ni ejecuta nuestro código, al contrario de lo que ocurre con lenguajes del lado servidor como PHP o ASP.

El uso de JavaScript se ha ido extendiendo y evolucionando a medida que la complejidad de las páginas lo requería. De este modo podemos encontrarnos con técnicas de desarrollo como:

- **DHTML** (*Dynamic HTML*): Mediante el uso de HTML, CSS y JavaScript esta técnica es capaz de crear efectos visuales y aumentar la interactividad con el usuario.
- **AJAX** (*Asynchronous JavaScript And XML*): Es una técnica capaz de solicitar datos adicionales al servidor en un segundo plano sin recargar o influir en la página que se está mostrando en ese momento. Lo normal es que la página se recargue por completo para poder mostrar la nueva información.

3.7. jQuery

jQuery es la librería JavaScript por excelencia, simplificando la programación de rutinas, manipulación del DOM e interacción con los elementos HTML en tu página. Es también utilizado para animaciones, solicitudes asíncronas mediante AJAX entre otros. Su autor original es John Resig, aunque como sucede con todas las librerías exitosas, actualmente recibe contribuciones de decenas de programadores. jQuery también ha sido programada de forma muy eficiente y su versión comprimida apenas ocupa 20 KB. Otra característica muy importante es el soporte para múltiples navegadores, lo cual para cualquier desarrollador web, reduce el tiempo de implementación.

Específicamente jQuery facilita:

- La búsqueda y manipulación de contenido en una página HTML.
- Trabajar con el modelo de eventos de los exploradores modernos.
- Y añadir efectos y transiciones sofisticadas que vemos en páginas modernas, como animaciones disparadas por eventos.

3.8. Json

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Para los humanos es sencillo leerlo y escribirlo, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 publicado en diciembre de 1999 [28]. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

La estructura de Json está constituida por dos estructuras perfectamente válidas en cualquier lenguaje de programación:

- Una colección de pares de nombre/valor.
- Una lista ordenada de valores.

Por otra parte, el valor de un par (clave, valor) puede ser un elemento simple (cadena de caracteres o número) o bien una estructura de datos. Esto permite representar estructuras de una complejidad arbitraria.

La especificación completa de este formato se puede consultar desde el sitio web de JSON [46].

3.9. Bootstrap

Es un es un framework front-end desarrollado y liberado por Twitter. Ha sido diseñado para simplificar el proceso de creación de diseños web. Para ello Bootstrap ofrece una serie de plantillas CSS y de ficheros JavaScript, los cuales permiten conseguir:

- Interfaces que funcionen de manera brillante en los navegadores actuales, y correcta en los no tan actuales.
- Un diseño que pueda ser visualizado de forma correcta en distintos dispositivos y a distintas escalas y resoluciones.
- Un diseño sólido basado en herramientas actuales y potentes como LESS o estándares como CSS3/HTML5

Bootstrap requiere de **jQuery** para funcionar. jQuery es una librería JavaScript de código abierto que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM(figura 3.3), manejo de eventos, desarrollar animaciones y agregar interacción con la técnica AJAX en páginas web.

Bootstrap incluye una rejilla o retícula fluida pensada para móviles y que cumple con el diseño web responsivo. Esta retícula crece hasta 12 columnas a medida que crece el tamaño de la pantalla del dispositivo. Bootstrap incluye clases CSS para manejar fácilmente la rejilla y hacer los diseños mas semánticos.

Bootstrap incluye todo tipo de elementos para hacer mas atractivas las páginas web. Estos elementos incluyen:

- Botones y grupos de botones

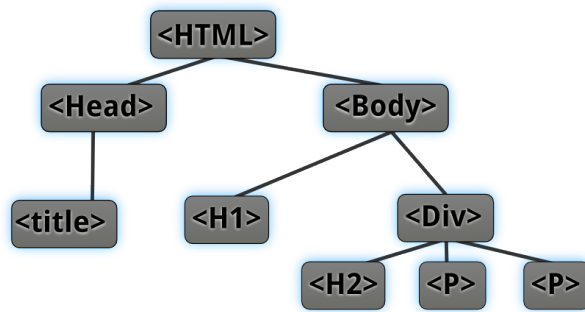


Figura 3.3: Representación del DOM

- Tablas
- Formularios
- Barras de navegación
- Iconos (glyphicons)
- Menús desplegables
- Imágenes en miniatura
- etc..

Bootstrap es libre, gratuito y puede descargarse los ficheros fuente así como consultar toda la documentación desde su página web [6].

Entre las técnicas de diseño centrado en el usuario en la fase de análisis se encuentran las entrevistas, los estudios de campo y las sesiones de grupo, que sirven para saber más sobre el usuario final y cómo interactúa con los productos. Estas mismas técnicas sirven también para saber más sobre las personas con discapacidad y los problemas de accesibilidad que presentan los productos.

4.1. Análisis de usuarios

En la siguiente sección realizamos un análisis exhaustivo de los distintos usuarios a los que va dirigida la aplicación. Comenzamos realizando el perfil de los usuarios objetivo de la aplicación. Posteriormente dividimos los usuarios objetivo en una serie de grupos... Finalmente se extraen los requisitos de interfaz de usuario a partir de este análisis.

Un componente esencial de la fase de análisis del diseño centrado en el usuario (DCU) es el análisis de usuario, que proporciona detalles sobre quién utiliza ese producto. El análisis de usuario identifica roles y define las características del usuario (como su nivel de conocimiento, experiencia y habilidad con productos similares; su entorno; frecuencia de uso; y dependiendo del tipo de producto, su hardware, software y tecnologías de apoyo que utiliza).

4.1.1. Usuarios objetivo

En el diseño centrado en el usuario, es importante tener muy claro como son las personas a las que va dirigida la aplicación. En este apartado, analizamos las características que tienen...

Se han encontrado dos tipos de usuarios. Los usuarios principales, que son los alumnos del centro. Y los usuarios secundarios, que son los docentes.

Perfil general de los usuarios primarios de la aplicación (Alumnos)

La tabla 4.1 muestra el perfil de usuario de los alumnos.

| Característica | Características del usuario |
|-----------------------------|---|
| Edad | Rango entre 6 y 21 años |
| Genero | Ambos |
| Limitaciones físicas | Puede tener desde ligeras a graves limitaciones físicas, intelectuales y/o sensoriales. |
| Educación | Pueden presentar necesidades educativas significativas según el grado y concurrencia de otras discapacidades. . |
| Uso de TIC | Están habituados a utilizar sistemas digitales en su día a día, pero siempre con supervisión. |
| Motivación | Necesitan una estimulación sensorial/basal que les ayude a conocerse a sí mismos y a relacionarse con su entorno... |
| Actitud | Será positiva si puede obtener el rendimiento esperado.. |

Tabla 4.1: Perfil general de los alumnos

Perfil general de los usuarios secundarios de la aplicación (Educadores)

La tabla 4.2 muestra el perfil de usuario de los educadores.

| Característica | Características del usuario |
|-----------------------------|--|
| Edad | Rango entre 18 y 67 años |
| Genero | Ambos |
| Limitaciones físicas | Puede ser totalmente capaz o tener algunas limitaciones físicas. |
| Educación | Tendrán una formación superior |
| Uso de TIC | Están habituados a utilizar sistemas digitales en su actividad profesional. |
| Motivación | Se trata de un usuario que necesita un apoyo para ejercer su actividad profesional de educador |
| Actitud | Será positiva si puede utilizar la aplicación en su actividad profesional con los niños |

Tabla 4.2: Perfil general de los educadores

4.1.2. Perfil (grupos) de los usuarios

Para realizar un análisis mas detallado, dividimos los distintos usuarios de la aplicación en una serie de grupos usando como diferenciadores fundamentales, la edad y las limitaciones físicas e intelectuales.

La tabla 4.3 muestra las características de los usuarios divididos por grupos. Se han elegido esos atributos para dividir los grupos de usuarios, ya que por un lado se tienen a los alumnos, diferenciados por sus limitaciones físicas y psicológicas, y en el otro lado, se tienen a los educadores, diferenciados por la edad, ya que la edad del educador marca sus conocimientos informáticos.

| Característica | Características del usuario por grupo | | |
|-----------------------------------|--|--|--|
| Edad | De 6 a 21 | De 18 a 45 | De 45 a 67 |
| Genero | Ambos | Ambos | Ambos |
| Limitaciones físicas | Puede tener discapacidad física en algún grado, incluyendo posibles limitaciones en el movimiento | Ninguna | Ninguna |
| Limitaciones intelectuales | Puede tener discapacidad intelectual en algún grado | Ninguna | Ninguna |
| Limitaciones sensoriales | Puede tener alguna discapacidad sensorial | Puede tener alguna deficiencia visual o auditiva muy leve | Puede tener alguna deficiencia visual o auditiva muy leve |
| Educación | Pueden presentar necesidades educativas significativas según el grado y concurrencia de otras discapacidades | Tendrán una formación superior | Tendrán una formación superior |
| Uso de TIC | Están habituados a utilizar sistemas digitales en su día a día, pero siempre con supervisión | Están habituados a utilizar sistemas digitales en su actividad profesional | Pueden tener dificultades en utilizar sistemas digitales en su actividad profesional |
| Motivación | Necesitan una estimulación sensorial/basal que les ayude a conocerse a sí mismos y a relacionarse con su entorno | Se trata de un usuario que necesita un apoyo para ejercer su actividad profesional de educador | Se trata de un usuario que necesita un apoyo para ejercer su actividad profesional de educador |
| Actitud | Será positiva si puede obtener el rendimiento esperado | Será positiva si puede utilizar la aplicación en su actividad profesional con los niños | Será positiva si puede utilizar la aplicación en su actividad profesional con los niños |

Tabla 4.3: Grupos de usuarios

4.1.3. Extracción de requisitos de la IU a partir del análisis de usuarios

La tabla 4.4 muestra los requisitos de interfaz de usuario e interacción que se han obtenido a partir del análisis de usuarios anterior.

4.1.4. Descripción de las personas

Describimos a Alba, Paloma y a Miguel, como los tres personajes representativos de cada uno de los grupos de usuarios identificados.

- **Alba** es una niña de 10 años que presenta una discapacidad física e intelectual. Es alumno del

| Característica | Características del usuario |
|-----------------------------------|---|
| Limitaciones visuales | Los elementos de la interfaz debe ajustara un tamaño adecuado para ser distinguidos correctamente por personas con limitaciones visuales |
| Limitaciones intelectuales | El funcionamiento ha de ser sencillo para que pueda ser entendido por personas con graves discapacidades intelectuales |
| Limitaciones físicas | Puede tener desde ligeras a graves limitaciones físicas e intelectuales |
| Educación | Ha de evitarse términos escritos. Utilizando en su lugar comunicación pictográfica |
| Uso de TIC | Como los usuarios pueden ser personas sin mucha experiencia en el mundo de las TIC, la interfaz debe proporcionar facilidad de aprendizaje y facilidad de uso |
| Distribución en pantalla | Los elementos que se utilicen tendrán que estar perfectamente diferenciados entre sí mismos y diferenciados del fondo que se utilice |

Tabla 4.4: Extracción de requisitos de la IU a partir del análisis de usuarios

centro Obregón. Aunque tiene que usar una silla de ruedas, puede manejar las manos con cierta soltura. Su parte favorita de la clase en el centro, es cuando le permiten "Jugar" con la tableta.

- **Paloma** tiene 27 años y es una educadora del centro Obregón. Como toda chica joven, tiene cierta soltura en el uso de las TIC, y en concreto, no tiene problemas al utilizar la Tableta Android ya que utiliza el mismo sistema operativo que su teléfono. Esta muy contenta de poder utilizar este tipo de tecnología con los alumnos, ya que estos se motivan mucho y trabajan mejor con este tipo de herramientas que con las herramientas tradicionales.
- **Miguel** tiene 53 años y lleva muchos años como educador del centro Obregón. Su teléfono móvil no es inteligente y no tiene ninguna tableta en casa. Debido a su desconocimiento con las tecnologías móviles, ha pedido ayuda a algún educador del centro mas joven para que le ayude en el manejo de las tablets.

4.2. Análisis de tareas

Tarea 1 - Seleccionar Minijuegos

Meta: El profesor debe seleccionar uno de los minijuegos propuestos.

Subtareas:

- Jugar a "Fuegos artificiales"
- Jugar a "Partículas"
- Jugar a "Estrellas"
- Jugar a "Trazos"

- Jugar a "Cantantes"
- Jugar a "Huevo"
- Jugar a "Dibujos animados"
- Jugar a "Canciones"

Tarea 2 - Configurar el tiempo de retardo de la aplicación

Meta: El monitor configurará a su gusto el tiempo que pasa desde que el alumno hace click hasta que el evento se dispara

Tarea 3 - Configurar la velocidad de la aplicación

Meta: El monitor configurará a su gusto la velocidad a la que aparecen los estímulos

Tarea 4 - Configurar el barrido

Meta: El monitor activará o desactivará a voluntad la funcionalidad de barrido en la aplicación

Tarea 5 - Configurar la velocidad del barrido

Meta: El monitor configurará a su gusto la velocidad a la que el barrido pasa de un elemento al siguiente

4.3. Modelo de dominio

El modelo de dominio es el artefacto más importante que se crea durante el análisis orientado a objetos. Es una visualización de los conceptos del dominio; es similar al modelo de información estático de las entidades del dominio. Es importante enfatizar que el modelo de análisis es una representación conceptual, correspondiente al problema y modelo de requisitos, en términos de clases de objetos. Cada una de estas clases contribuye de manera especial a lograr la arquitectura deseada.

4.4. Requisitos funcionales

Los principales requisitos funcionales son:

- **RF1-Minijuegos:** La aplicación consistirá en ocho minijuegos diferentes
- **RF2-Configuración:** Los educadores deben poder configurar los ajustes de la aplicación
- **RF3-Desbloqueo:** Los minijuegos podrán disponer de un elemento que evitará la pulsación no deseada de botones

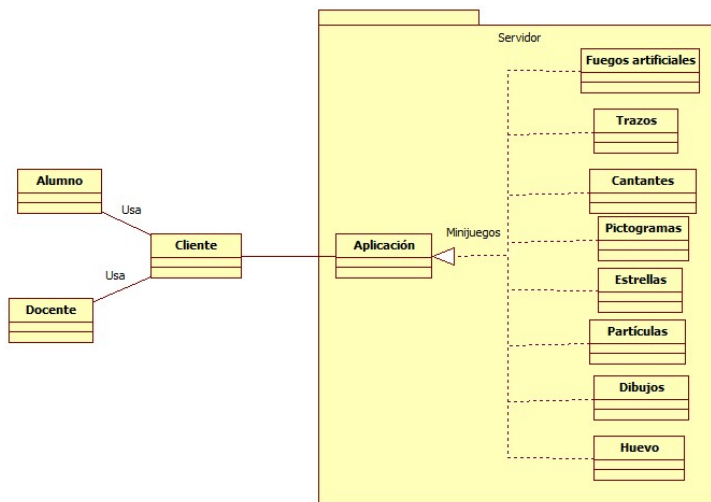


Figura 4.1: Modelo de dominio

- **RF4-Pantalla completa:** Los vídeos deberá poder reproducirse en pantalla completa
- **RF5-Añadir nuevos elementos:** Los usuarios deben ser capaces de añadir nuevos elementos multimedia al sistema, pasando a ser parte de la aplicación en usos futuros
- **RF6-Reinicio:** Los minijuegos podrán reiniciarse un numero ilimitado de veces desde el propio minijuego.
- **RF7-Juego fuegos artificiales - Número de elementos:** El número de elementos en pantalla del minijuego de los fuegos artificiales será configurable
- **RF8-Juego estrellas - Número de elementos:** El número de elementos en pantalla del minijuego de las estrellas será configurable

4.5. Requisitos de usabilidad

- **RU1-Consistencia:** El aspecto de la aplicación debe ser consistente en toda la aplicación
- **RU2-Comprensión:** La aplicación deberá utilizar mensajes sencillos y claros

4.5.1. Requisitos de interfaz de usuario

- **IU1-Colorido:** El aspecto gráfico de la interfaz de la aplicación deberá ser colorido
- **IU2-Contraste:** La aplicación deberá tener un alto grado de contraste entre los elementos en pantalla y el fondo
- **IU3-Tamaño de los elementos:** Los objetos mostrados en pantalla deberán ser lo mas grandes posibles

- **IU4-Pictogramas:** Los elementos mostrados en pantalla deberán ser o los elementos proporcionados por el cliente o símbolos ARASAC
- **IU5-Aspecto menú inicio:** El menú de inicio organizará los elementos mostrados en pantalla en formato de rejilla
- **IU6-Orientación:** El juego se mostrará únicamente en posición horizontal, y variaciones en la orientación de la tableta no modificará la orientación de la pantalla
- **IU7-Tipografía:** La tipografía de los textos debe ser "Escolar 2"

Descripción del modelo

Para la descripción del modelo se utiliza UML (Unified Modeling Language), el cual nos permitirá visualizar, especificar, construir y documentar el sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como, procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

4.6. Especificación de casos de uso

4.6.1. Modelo de casos de uso

El modelo de casos de uso describe la funcionalidad propuesta del sistema. Un caso de uso representa una unidad discreta de interacción entre un usuario y el sistema. Los diagramas que se obtienen en esta fase, permitirán visualizar el modo en que un actor opera con un sistema en desarrollo, además de la forma, el tipo y orden en que los elementos interactúan. Estos diagramas son especialmente importantes para organizar y modelar el comportamiento de un sistema.

Diagrama del modelo de casos de uso

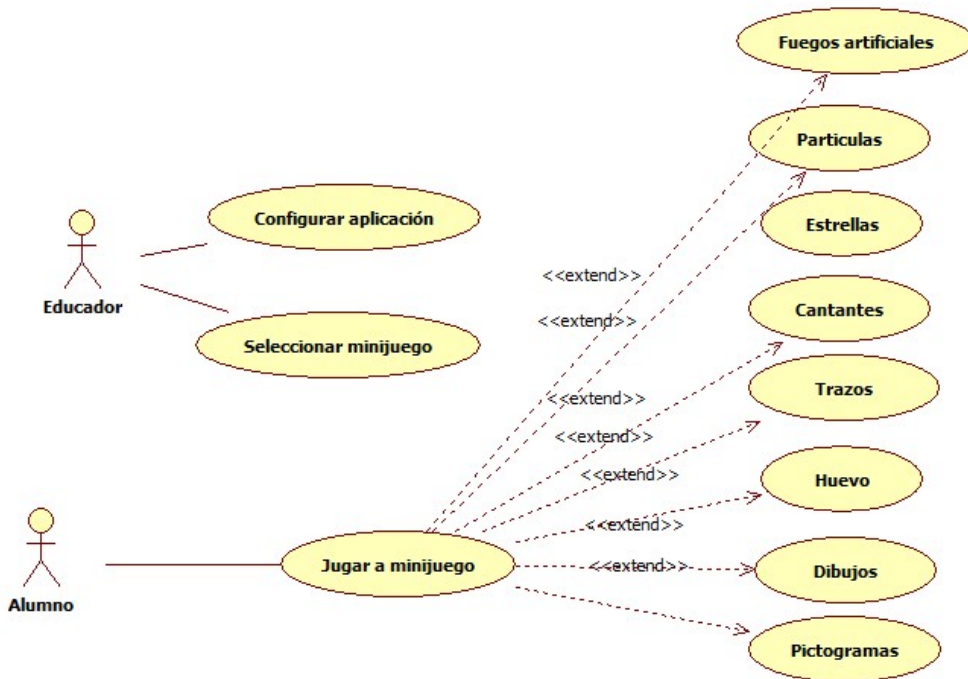


Figura 4.2: Diagrama de casos de uso

4.6.2. CU-1: Configurar aplicación

| CU-1 | Configurar aplicación |
|-------------------------|---|
| Actor | Educador |
| Función | Se configuran algunos aspectos del sistema |
| Precondiciones | El actor debe de estar en la pantalla principal de la aplicación |
| Secuencia normal | 1 - El educador accede a la pantalla de configuración del sistema pulsando en el botón de configuración. 2 - El sistema muestra los distintos aspectos de la aplicación que el actor puede configurar. |
| Excepción 1 | 1 - En cualquier momento a partir del paso 2 el educador puede cancelar la gestión y el sistema le devuelve al menú principal. |
| Comentarios | Con este caso de uso, el actor es capaz de configurar algunos aspectos del sistema. |

Tabla 4.5: CU-1: Configurar aplicación

4.6.3. CU-2: Seleccionar minijuego

| CU-2 | Seleccionar minijuego |
|-------------------------|---|
| Actor | Educador |
| Función | Permitir o impedir la pulsación accidental de los botones atrás y reinicio de la aplicación |
| Precondiciones | El actor debe de estar en la pantalla principal de la aplicación |
| Secuencia normal | 1 - El sistema muestra los minijuegos disponibles 2 - El educador selecciona uno de los minijuegos 3 - El sistema lanza el minijuego seleccionado |
| Excepción 1 | 1 - Si alguno de los minijuegos no esta disponible en ese momento, el sistema mostrará un mensaje indicándolo y volverá al punto 1. |
| Comentarios | Con este caso de uso, el actor accede a los minijuegos disponibles de la aplicación |

Tabla 4.6: CU-2: Seleccionar minijuego

4.6.4. CU-3: Jugar a minijuego

| CU-3 | Jugar a minijuego |
|-------------------------|--|
| Actor | Alumno |
| Función | El sistema permite ejecutar los distintos minijuegos de los que se compone la aplicación |
| Precondiciones | El actor debe de estar en la pantalla principal de la aplicación |
| Secuencia normal | 1 - El sistema ejecuta el minijuego que el educador ha seleccionado en el CU-2 |
| | |
| Comentarios | Con este caso de uso, el sistema permite jugar a los minijuegos disponibles en la aplicación |

Tabla 4.7: CU-3: Jugar a minijuego

CU-3.1: Fuegos artificiales

| CU-3.1 | Fuegos artificiales |
|-------------------------|--|
| Actor | Alumno |
| Función | El sistema muestra fuegos artificiales y cohetes, que explotan al ser tocados |
| Precondiciones | El actor debe de estar en el minijuego "Fuegos artificiales" |
| Secuencia normal | 1 - El sistema muestra una serie de fuegos artificiales y cohetes 2 - El actor pulsa sobre alguno de los fuegos artificiales o cohete 3 - EL fuego artificial o cohete explota |
| | |
| Excepción 1 | 1 - En cualquier momento el actor puede cancelar el caso de uso, volviendo al menú principal |
| Comentarios | Con este caso de uso, el actor juega al minijuego fuegos artificiales |

Tabla 4.8: CU-3.1: Fuegos artificiales

CU-3.2: Partículas

| CU-3.2 | Partículas |
|-------------------------|---|
| Actor | Alumno |
| Función | El sistema muestra partículas que vuelan zumbando desde la punta del dedo del actor |
| Precondiciones | El actor debe de estar en el minijuego "Partículas" |
| Secuencia normal | 1 - El sistema muestra algún indicador que llame a pulsar con el dedo la pantalla 2 - El actor pulsa la pantalla con el dedo 3 - Aparecen partículas zumbando desde la punta del dedo |
| Excepción 1 | 1 - En cualquier momento el actor puede cancelar el caso de uso, volviendo al menú principal |
| Comentarios | Con este caso de uso, el actor juega al minijuego partículas |

Tabla 4.9: CU-3.2: Partículas**CU-3.3: Estrellas**

| CU-3.3 | Estrellas |
|-------------------------|---|
| Actor | Alumno |
| Función | El sistema muestra una pantalla con estrellas que al tocarlas se iluminan y pueden ser desplazadas con el dedo |
| Precondiciones | El actor debe de estar en el minijuego "Estrellas" |
| Secuencia normal | 1 - El sistema muestra una serie de estrellas bajo un fondo estrellado |
| Alternativa 1 | 2 - El actor pulsa sobre alguna de las estrellas 3 - La estrella se ilumina |
| Alternativa 2 | 2 - El actor pulsa con el dedo una estrella y lo arrastra por la pantalla 3 - La estrella se mueve por la pantalla siguiendo al dedo |
| Excepción 1 | 1 - En cualquier momento el actor puede cancelar el caso de uso, volviendo al menú principal |
| Comentarios | Con este caso de uso, el actor juega al minijuego de las estrellas |

Tabla 4.10: CU-3.3: Estrellas

CU-3.4: Trazos

| CU-3.4 | Trazos |
|-------------------------|---|
| Actor | Alumno |
| Función | El sistema muestra trazos de colores cuando el actor pasa el dedo por la pantalla |
| Precondiciones | El actor debe de estar en el minijuego "Trazos" |
| Secuencia normal | 1 - El sistema muestra un lienzo en blanco 2 - El actor pulsa sobre la pantalla con el dedo 3 - Se muestran trazos por donde el actor ha tocado la pantalla |
| Excepción 1 | 1 - En cualquier momento el actor puede cambiar el color del trazo pulsando sobre la opción correspondiente |
| Excepción 2 | 1 - En cualquier momento el actor puede cancelar el caso de uso, volviendo al menú principal |
| Comentarios | Con este caso de uso, el actor juega al minijuego de los trazos |

Tabla 4.11: CU-3.4: Trazos

CU-3.5: Cantantes

| CU-3.5 | Cantantes |
|-------------------------|--|
| Actor | Alumno |
| Función | El sistema muestra la foto de un cantante, que al ser tocada reproduce un vídeo de su música |
| Precondiciones | El actor debe de estar en el minijuego "Cantantes" |
| Secuencia normal | 1 - El sistema muestra la imagen de uno o mas cantantes 2 - El actor pulsa sobre la foto de un cantante 3 - El sistema muestra un vídeo de la música de ese cantante |
| Excepción 1 | 1 - En cualquier momento el actor puede cancelar el caso de uso, volviendo al menú principal |
| Comentarios | El sistema puede mostrar la imagen de uno o mas cantantes |

Tabla 4.12: CU-3.5: Cantantes

CU-3.6: Huevo

| CU-3.6 | Huevo |
|-------------------------|---|
| Actor | Alumno |
| Función | El sistema muestra la imagen de un huevo, que al tocarlo se rescrebaja haciendo un ruido y sale un pollito |
| Precondiciones | El actor debe de estar en el minijuego "huevo" |
| Secuencia normal | 1 - El sistema muestra la imagen de un huevo 2 - El actor pulsa sobre la imagen de huevo 3 - El huevo se rescrebaja haciendo un ruido y de el sale un pollito |
| Excepción 1 | 1 - En cualquier momento el actor puede cancelar el caso de uso, volviendo al menú principal |
| Comentarios | Con este caso de uso, el actor juega al minijuego de del huevo |

Tabla 4.13: CU-3.6: Huevo**CU-3.7: Dibujos**

| CU-3.7 | Dibujos |
|-------------------------|---|
| Actor | Alumno |
| Función | El sistema muestra una serie de dibujos animados, que al pulsarlo reproduce un episodio |
| Precondiciones | El actor debe de estar en el minijuego "Dibujos" |
| Secuencia normal | 1 - El sistema muestra algunos personajes de dibujos animados 2 - El actor pulsa alguno de los personajes de dibujos animados 3 - El sistema reproduce un episodio de la serie de dibujos animados que ha seleccionado el actor |
| Excepción 1 | 1 - En cualquier momento el actor puede cancelar el caso de uso, volviendo al menú principal |
| Comentarios | Con este caso de uso, el actor puede seleccionar la serie de dibujos animados que prefiera ver |

Tabla 4.14: CU-3.7: Dibujos

CU-3.8: Pictogramas

| CU-3.8 | Pictogramas |
|-------------------------|--|
| Actor | Alumno |
| Función | El sistema muestra un pictograma relacionado con una canción, que al ser pulsado se oye o ve el vídeo de la canción |
| Precondiciones | El actor debe de estar en el minijuego "Pictogramas" |
| Secuencia normal | 1 - El sistema muestra un pictograma 2 - El actor pulsa sobre un pictograma 3 - El sistema reproduce el audio o vídeo de la canción relacionada con el pictograma seleccionado |
| Excepción 1 | 1 - En cualquier momento el actor puede cancelar el caso de uso, volviendo al menú principal |
| Comentarios | El sistema puede mostrar uno o varios pictogramas |

Tabla 4.15: CU-3.8: Pictogramas

DCU: Prototipo inicial

En este capítulo realizamos y evaluamos un primer prototipo no funcional. El objetivo consiste en que el cliente de su opinión sobre el prototipo mostrado.

5.1. Introducción

5.1.1. ¿Qué es un prototipo?

Un prototipo puede ser un esquema en papel de una pantalla o pantallas, una imagen electrónica, una simulación en vídeo de una tarea, una representación en 3D de una estación de trabajo entre otras cosas, de hecho un prototipo puede ser cualquier cosa desde un storyboard basado en papel hasta una compleja pieza de software. De hecho, según Jenny Preece en su libro "Interaction Design: Beyond Human-Computer Interaction", *un prototipo es una representación limitada de un diseño y que permite a los usuarios interactuar con el y explorar sus posibilidades*[37]. El prototipo es una herramienta muy útil para hacer participar al usuario en el desarrollo y poder evaluar el producto ya en las primeras fases del desarrollo.

5.1.2. Tipos de prototipos

Prototipo de papel

Este tipo de prototipo se basa en la utilización de papel, tijeras, lápiz o instrumentos que se puedan utilizar para describir un diseño en un papel. Este sistema nos permite una gran velocidad y flexibilidad.

Para poder simular las diferentes interacciones que vamos a realizar con el sistema, realizaremos una hoja para cada uno de los diferentes escenarios que vamos a tener como resultado de las diferentes posibles interacciones que podemos realizar. Apilaremos estas hojas que nos permitirán simular la aplicación.

Para utilizar el prototipo de papel nos situaremos en un escenario de uso de futuro en el que el diseñador actúa como coordinador. El prototipo será analizado por un posible usuario e intentará realizar algunas de las tareas que se pretende diseñar. En voz alta se irán realizando las interacciones y le iremos cambiando las hojas de papel en función de las interacciones que vaya realizando.

El coste es muy reducido, necesitando únicamente los recursos humanos dedicados a la realización del prototipo. Los cambios se pueden realizar muy rápidamente y sobre la marcha. Si el diseño no funciona se puede reescribir las hojas erróneas o rediseñarlas y volver a probar la tarea a realizar.

Los usuarios o los actores se sienten más cómodos para poder realizar críticas al diseño debido a la sencillez del mismo por lo que no se sienten cohibidos a dar sus opiniones.

Vídeo

El vídeo nos permite el rodaje de un escenario en el que podemos realizar manipulaciones durante el postproceso para simular algunas características del diseño de las que todavía no disponemos. El prototipo en vídeo puede ser muy útil en el diseño de interfaces multimodales en el que por ejemplo se realiza una interacción por voz o en el diseño de escenarios futuros de los que todavía no se dispone de la tecnología. El vídeo se visiona después por el equipo de desarrollo y posibles usuarios y puede ser evaluado por ejemplo con las técnicas de pensar en voz alta.

Simulaciones

Algo de funcionalidad tiene que ser incluida en el prototipo para demostrar el trabajo que la aplicación tiene que realizar. Los prototipos presentados hasta ahora no son suficientes para este propósito, Para poder realizar este trabajo parte o toda la funcionalidad del sistema tiene que ser simulada por el equipo de diseño. Añadiendo un soporte de programación para simular permite al diseñador construir objetos interactivos textuales y gráficos que añaden cierto comportamiento a estos objetos que simulan las funcionalidades del sistema. Una vez construida la simulación se puede evaluar y cambiar en función de los resultados de la evaluación.

Prototipo de software

En este caso se pueden realizar diferentes tipos de prototipos utilizando las herramientas de desarrollo. Tenemos las siguientes opciones:

- **Maqueta para tirar:** Es un tipo de prototipo parecido al de papel por ejemplo, en que se sirve para realizar una evaluación con el usuario y pos-teriormente se desecha.
- **Incremental:** El producto final se construye como componentes separados. Cada vez se va probando uno y finalmente se realiza una prueba final.
- **Evolutivo:** En este caso el prototipo no es eliminado y se utiliza como base para una próxima iteración en el diseño.

5.2. Entrevistas con usuarios

Las entrevistas con usuarios son una poderosa herramienta cualitativa, pero no para evaluar la usabilidad de un diseño, sino para descubrir deseos, motivaciones, valores y experiencias de nuestros usuarios

Durante estas entrevistas, el entrevistador debe mostrarse neutral y no dirigir o condicionar las respuestas del entrevistado. Lo que pretendemos es descubrir información que nos oriente en el diseño, no confirmar nuestras propias creencias sobre cómo son los usuarios.

Una variante interesante de las entrevistas, son los focus group (en español grupos focales o sesiones de grupo), en las que un moderador entrevista de forma conjunta a un grupo de usuarios, y donde la interacción entre los participantes nos ofrece información adicional sobre problemas, experiencias o deseos compartidos.

5.3. Evaluación del prototipo de baja definición

A continuación les presentamos un prototipo de la aplicación de Causa-efecto. Para este prototipo se han elegido colores alegres, para motivar al niño a usar la aplicación.

Para los textos se ha utilizado el color negro para resaltar así de los colores de fondo. La evaluación sirve para conocer las opiniones de los educadores del centro del prototipo propuesto. Los educadores examinarán este prototipo ya que son ellos los que mejor conocen las posibles reacciones de los alumnos ante los estímulos visuales, táctiles y auditivos que proporcionará la aplicación, y determinarán si la temática, distribución en pantalla o colores elegidos son adecuados. Cabe destacar que lo ahora expuesto no es más que un prototipo pensado para conocer las modificaciones que los educadores quieran que se lleven a cabo. A continuación mostramos cada una de las pantallas de la aplicación. Y se realizarán algunas preguntas a los educadores para conocer sus impresiones sobre el prototipo. Además, los educadores tienen la libertad de aportar libremente los comentarios que consideren oportunos con tal de ayudar a mejorar la aplicación.

5.3.1. Menú principal



Figura 5.1: Prototipo Low: Menú Principal

Explicación

Este es el menú principal de la aplicación (Figura 5.1). En esta pantalla se muestran cada uno de los juegos disponibles en la aplicación. La distribución de los juegos disponibles se realiza mediante una rejilla en el centro de la pantalla. Cada juego está representado por el título del juego y por una imagen representativa de este. En la parte superior aparece el botón para salir de la aplicación representado con una cruz, y el botón para acceder al menú de configuración está representado por un engranaje. Se utilizan botones del mismo color, para no llamar la atención del alumno sobre estos, ya que toda la atención se debe centrar en los minijuegos.

Preguntas

- **¿La distribución de los elementos en pantalla es adecuada?**
Si
- **¿El tamaño y tipografía de los textos es adecuada?**
La letra debería ser "Escolar 2"
- **¿Se entiende correctamente la funcionalidad de los botones?**
Si
- **¿El color de los botones es adecuado?**
Si
- **¿Los nombres de los juegos y su imagen representativa son suficientemente descriptivos?**
La imagen del minijuego estrellas, debería ser una estrella grande. La imagen del minijuego canciones, debería tener el símbolo Arasac de Musica. Un radiocasete, o una nota musical

5.3.2. Juego de los fuegos artificiales



Figura 5.2: Prototipo Low: Fuegos Artificiales

Explicación

Este minijuego trata de que cuando se aprieten los fuegos artificiales que aparecen en pantalla, estos zumben y explotan. Inicialmente los menús se encuentran desbloqueados. Pulsando el icono del candado se bloquean, haciendo desaparecer los botones, impidiendo su pulsación accidental.

Preguntas

- **¿El tamaño de los cohetes y fuegos artificiales en pantalla es correcto?**
Si



Figura 5.3: Prototipo Low: Fuegos Artificiales

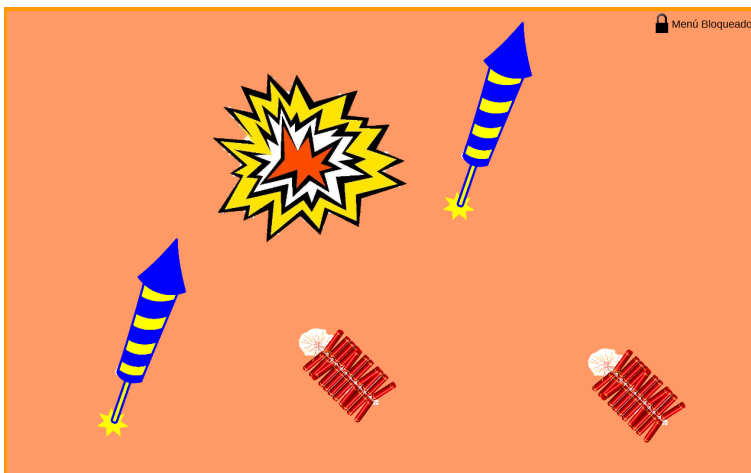


Figura 5.4: Prototipo Low: Fuegos Artificiales

- **¿ La cantidad de elementos en pantalla es adecuada? ¿ Es necesario poner más elementos?**
Está bien, pero la cantidad de elementos que explotan debería ser configurable.
- **¿ Se entiende bien en qué consiste el juego?**
Si
- **¿ Los dibujos son los adecuados?**
Si. Son muy coloridos.
- **¿ Se entiende la funcionalidad del botón de bloqueo/desbloqueo?**
Si. Aunque debería estar bloqueado por defecto.

5.3.3. Juego de las partículas

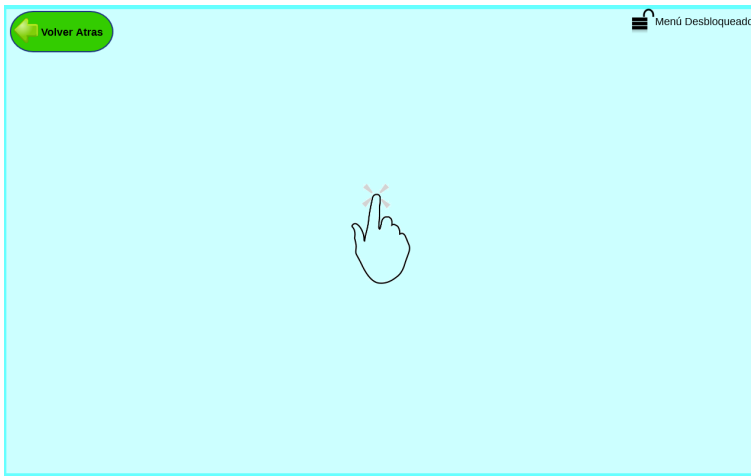


Figura 5.5: Prototipo Low: Juego de las partículas

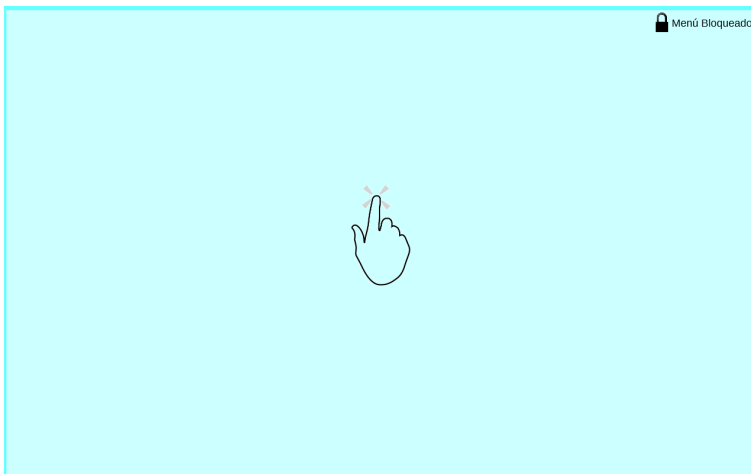


Figura 5.6: Prototipo Low: Juego de las partículas

Explicación

Este minijuego consiste en que cuando se pulse con el dedo la pantalla, se vean partículas zumbando desde la punta del dedo.

Preguntas

- ¿El fondo claro elegido es correcto?
Si



Figura 5.7: Prototipo Low: Juego de las partículas

- **¿La imagen del dedo pulsando la pantalla es representativo de lo que hay que hacer en el juego?**
Si, pero sería conveniente que la mano estuviese rellena de color para que destacase sobre el fondo.
- **¿El tamaño de las chispas es el adecuado?**
No. Deberían ser mucho mas grandes.

5.3.4. Juego de las estrellas

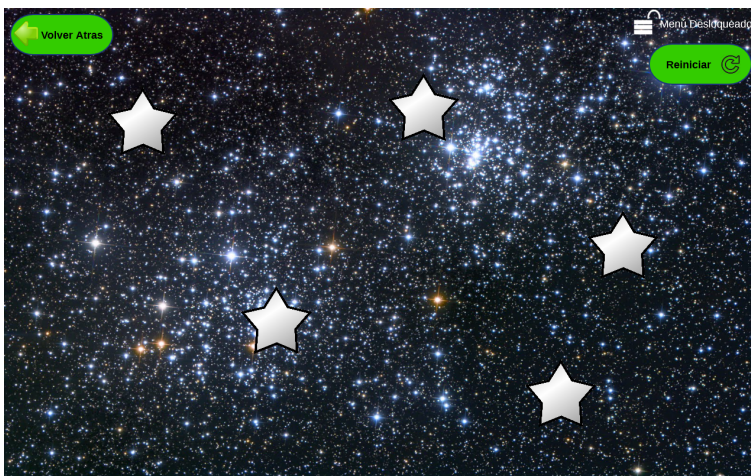


Figura 5.8: Prototipo Low: Juego de las estrellas



Figura 5.9: Prototipo Low: Juego de las estrellas



Figura 5.10: Prototipo Low: Juego de las estrellas

Explicación

Este juego consiste en que aparecen unas estrellas plateadas sobre un fondo estrellado. Al pulsar cada una de las estrellas esta se ilumina.

Preguntas

- **¿El tamaño de las estrellas es el adecuado?**
No. Deberían ser más grandes.
- **El fondo estrellado es adecuado?**
Si

- ¿Se entiende bien el funcionamiento del juego?
Si
- ¿El botón de bloqueo/desbloqueo destaca lo suficiente del fondo?
Si

5.3.5. Cantantes



Figura 5.11: Prototipo Low: Cantantes



Figura 5.12: Prototipo Low: Cantantes

Explicación

Este juego consiste en que cuando se toque la foto de un cantante, se reproduzca un vídeo de su música.

El botón de bloqueo/desbloqueo es de color blanco para que destaque bien sobre el fondo negro.

Preguntas

- **¿Se entiende bien el funcionamiento del juego?**
Si
- **¿El número de cantantes es adecuado?**
Si

5.3.6. Juego de los trazos



Figura 5.13: Prototipo Low: Juego de los trazos

Explicación

Este juego consiste en un lienzo en blanco con una serie de botones de colores. Cuando el niño pulsa sobre un color, este es seleccionado, y pintará el lienzo siguiendo los trazos llevados a cabo por el alumno. El botón reiniciar, limpiará la pantalla dejando el lienzo en blanco de nuevo.

Preguntas

- **¿Se entiende bien el funcionamiento del juego?**
Si
- **¿Los colores disponibles para pintar son los adecuados? ¿es necesario cambiar alguno? ¿su número es suficiente o se necesitan más colores?**

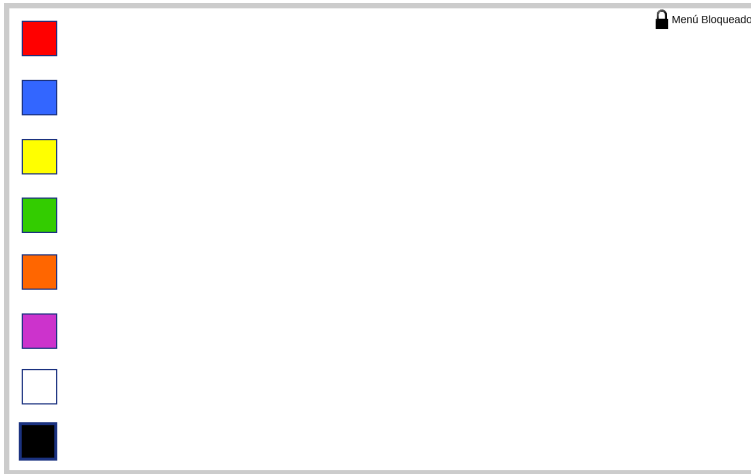


Figura 5.14: Prototipo Low: Juego de los trazos

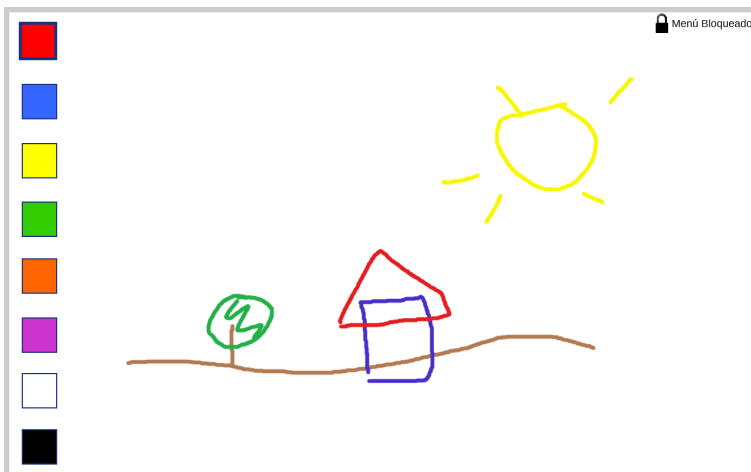


Figura 5.15: Prototipo Low: Juego de los trazos

Menos colores. Basta con los 4 colores básicos (rojo, azul, amarillo y verde) y el blanco representado por una goma de borrar al otro extremo de la pantalla.

- **¿Se diferencia adecuadamente que color está seleccionado?**
No. Mejor si fuera un poco mas grande.
- **¿El grosor de la línea es la adecuada?**
Si
- **¿Un único grosor de línea está bien?**
Si
- **¿El tamaño de la paleta de colores es la adecuada?**

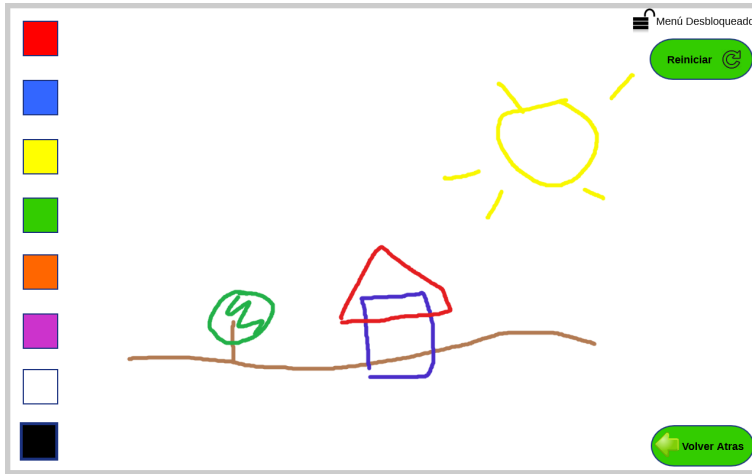


Figura 5.16: Prototipo Low: Juego de los trazos

No. Más grande.
Iconos circulares en lugar de rectangulares.

5.3.7. Juego del huevo

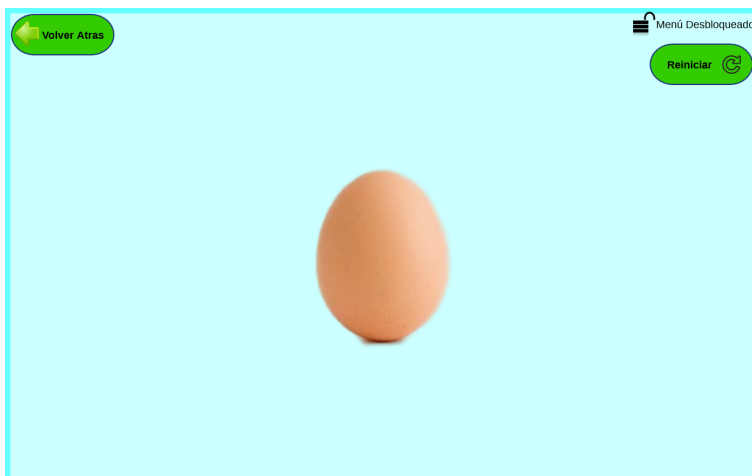


Figura 5.17: Prototipo Low: Juego del huevo

Explicación

Este juego consiste en que aparece en pantalla un huevo, y cuando el alumno lo pulse, se resquebraja haciendo un ruido y de él sale un pollito.

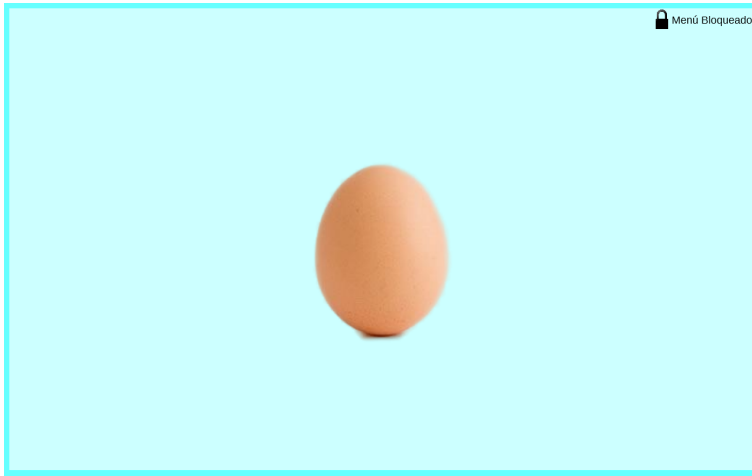


Figura 5.18: Prototipo Low: Juego del huevo

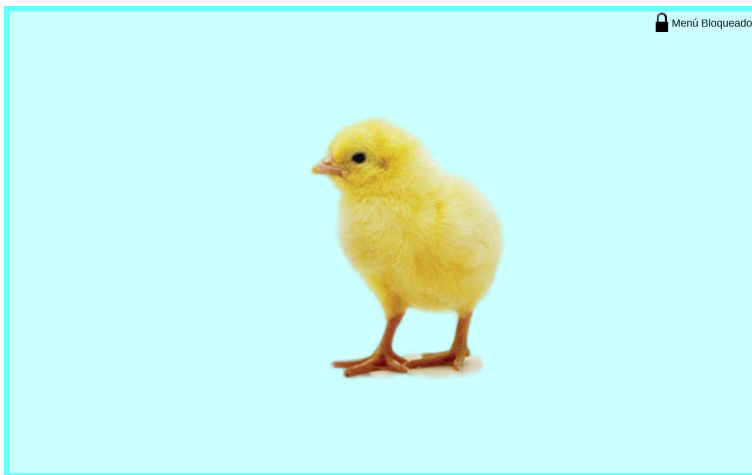


Figura 5.19: Prototipo Low: Juego del huevo

Preguntas

- **¿El tamaño del huevo y del pollito es el adecuado?**
No. Debe ser un poco más grande.
- **¿El área a pulsar por el alumno, debe ser solo el objeto en cuestión, o toda la pantalla?**
Un poco más grande que la imagen.
- **¿Es correcto que para poder reiniciar el juego se deba desbloquear la pantalla?**
Si

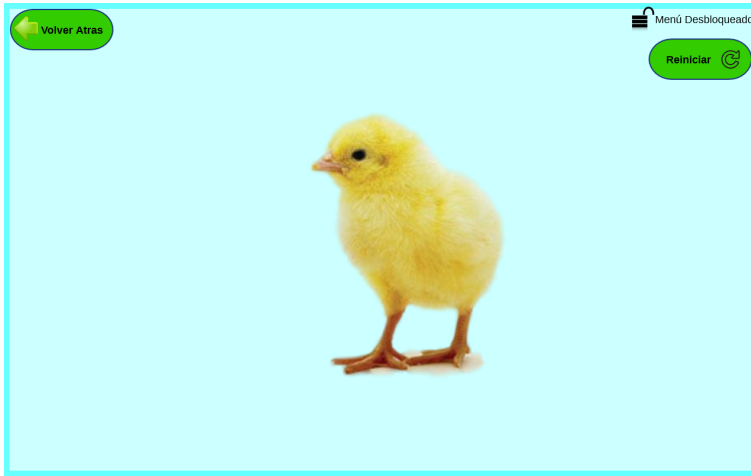


Figura 5.20: Prototipo Low: Juego del huevo



Figura 5.21: Prototipo Low: Dibujos animados

5.3.8. Dibujos animados

Explicación

En la pantalla aparecen las imágenes de distintas series de dibujos animadas famosas. Al pulsar en alguna de ellas, se reproduce un episodio de la serie seleccionada.

Preguntas

- **¿El tamaño de las series es el adecuado?**
Si
- **¿La cantidad de series de dibujos es suficiente o son necesarias más?**



Figura 5.22: Prototipo Low: Dibujos animados



Figura 5.23: Prototipo Low: Dibujos animados

Si, es suficiente.

5.3.9. Pictogramas y canciones

Explicación

En la pantalla aparece un pictograma asociado a una determinada canción. Al pulsar sobre el pictograma, se reproduce el audio de esa canción.

Preguntas

- ¿El formato de juego escogido es el adecuado?
Si

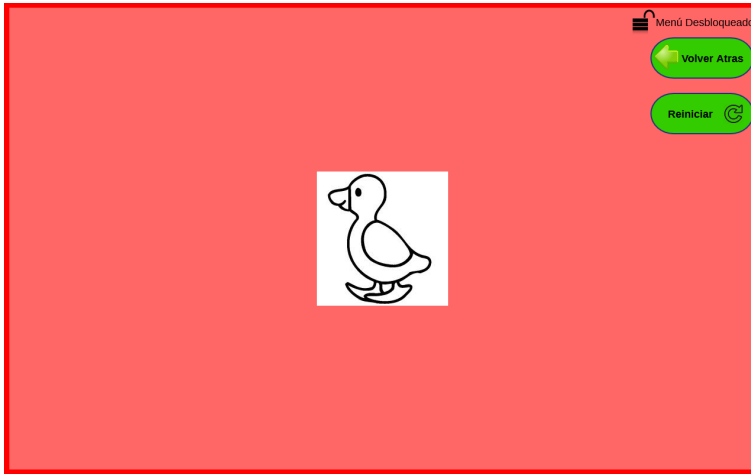


Figura 5.24: Prototipo Low: Pictogramas y canciones

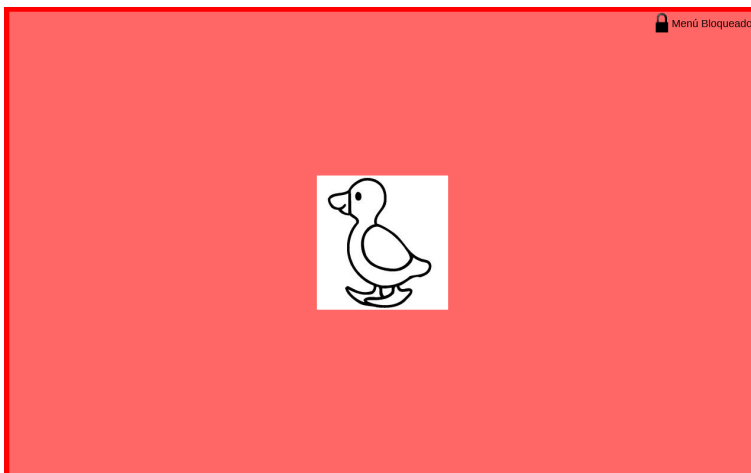


Figura 5.25: Prototipo Low: Pictogramas y canciones

- **¿Se entiende bien en qué consiste el juego?**
Si
- **¿El tamaño del pictograma es el adecuado?**
No. Debe ser mas grande

5.3.10. Comentarios adicionales realizados por los evaluados

- El vídeo de los minijuego de los cantantes, deberían reproducirse automáticamente
- El minijuego de los dibujos animados debería reproducir en pantalla completa directamente
- El minijuego de los pictogramas debería reproducir en pantalla completa directamente



Figura 5.26: Prototipo Low: Pictogramas y canciones

- El bloqueo/desbloqueo debería hacerse mediante una doble pulsación, ya que para los alumnos es muy difícil pulsar dos veces
- Los evaluados sugieren poder configurar multitud de parámetros de la aplicación, tales como:
 - Las imágenes representativas de la aplicación en el menú principal así como de su leyenda
 - Añadir imágenes o estímulos propios a la aplicación
 - Número de estímulos que aparecen en las aplicaciones

5.3.11. Conclusiones de la evaluación del prototipo de baja definición

La realización y evaluación de un prototipo de baja definición, ha permitido contrastar la idea inicial del desarrollador, con las necesidades del cliente. Todo ello sin gastar tiempo y esfuerzos innecesarios.

Adicionalmente, las propuestas que son factibles de realizar en el tiempo asignado son añadidas a los requisitos del proyecto.

El diseño de la aplicación, es la fase mas importante en el desarrollo de cualquier software. Si realizamos un mal diseño iremos arrastrando los problemas que esto genera a todas las fases del desarrollo, con el resultado de obtener software de mala calidad.

6.1. El modelo de capas

El modelo de capas es una técnica software para separar las diferentes partes de la aplicación, con el objetivo de mejorar su rendimiento, mantenimiento y sus funciones. Esta separación de las diferentes partes hace que los cambios en cualquiera de las capas no afecten o afecten poco a las otras capas en que está dividida la aplicación.

Las tres capas serían las siguientes:

- **Capa de presentación:** Es la capa que ve el usuario, presenta el sistema, captura la información y la presenta la información al usuario en un mismo proceso. Esta capa se comunica únicamente con la capa de negocio.
- **Capa de negocio o lógica:** Se reciben las peticiones del usuario y se envían las respuestas tras el proceso. En esta capa se establecen todas las reglas que deben cumplirse. Se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, también se comunica con la capa de persistencia o de datos para solicitar al gestor de la base de datos para recuperar, modificar o insertar datos en la base de datos.
- **Capa de persistencia o de datos:** Es donde residen los datos y la encargada de acceder a ellos. Esta formada por uno o mas gestores de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación desde la capa de negocio.

6.2. El patrón modelo, vista, controlador

Dentro del modelo de arquitectura en tres capas uno de los patrones de diseño de software más utilizado es el patrón MVC (Modelo, Vista, Controlador).

El patrón de diseño Modelo-Vista-Controlador (MVC) fue descrito por primera vez en el año 1979 por Trygve Reenskaug, de los laboratorios de I+D de Xerox. Su característica primordial es la separación de la aplicación en tres capas distintas, el Modelo, la Vista y el Controlador.

Así, el MVC es una guía para el diseño de arquitecturas de aplicaciones que mantienen una gran interactividad con los usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información y el tercero es un conjunto de controladores que procesan las peticiones de los usuarios y controla el flujo de ejecución del sistema.

La figura 6.1 muestra un diseño básico de la interacción entre las tres capas.

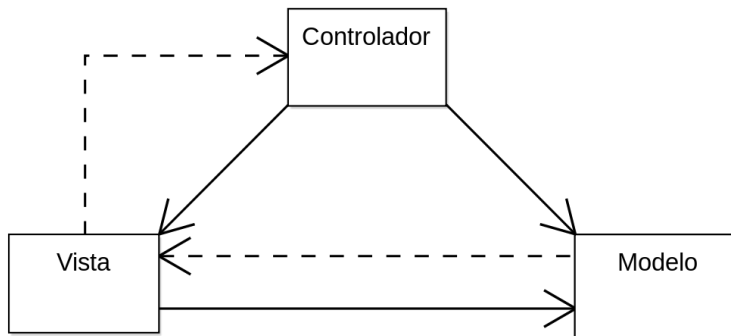


Figura 6.1: Diagrama del patrón MVC

Por tanto el MVC divide las aplicaciones en tres niveles de abstracción:

- **Modelo:** Representa la lógica de negocios. Es el encargado de acceder de forma directa a los datos actuando como "intermediario" con la persistencia. En una aplicación web, el modelo sería la información almacenada en una base de datos o JSON.
- **Vista:** Es la encargada de mostrar la información al usuario de forma gráfica y "humanamente legible". En una aplicación web, la vista sería la página HTML.
- **Controlador:** Es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario, de forma "humanamente legible". En una aplicación web, el controlador sería el código que obtiene datos dinámicamente y genera el contenido HTML.

La secuencia de funcionamiento de este patrón de diseño podría resumirse de la siguiente manera:

1. El usuario realiza una petición.
2. El controlador captura el evento.
3. Hace la llamada al modelo/modelos correspondientes efectuando las modificaciones pertinentes sobre el modelo
4. El modelo será el encargado de interactuar con la persistencia, ya sea en forma directa o con una capa de abstracción para ello y retornará esta información al controlador.
5. El controlador recibe la información y la envía a la vista.

- La vista, procesa esta información, creando una capa de abstracción para la lógica (quien se encargará de procesar los datos) y otra para el diseño de la interfaz gráfica o GUI. La lógica de la vista, una vez procesados los datos, los "acomodará" en base al diseño de la GUI y los entregará al usuario de forma "humanamente legible".

6.3. Elección del framework

Uno de los requisitos de este proyecto, es que la aplicación funcione bajo el sistema operativo Android. Sin embargo ese requisito no excluye que se pueda desarrollar también para otras plataformas. Por ello se ha decidido que la aplicación sea lo mas abierta posible, dando la posibilidad de usarlo en sistemas operativos diferentes e incluso mediante un navegador web.

Para conseguirlo, se ha decidido diseñar e implementar la aplicación utilizando las tecnologías web HTML5 Canvas, CSS3 y JavaScript, y utilizar el framework **PhoneGap** para generar las aplicaciones móviles.

PhoneGap

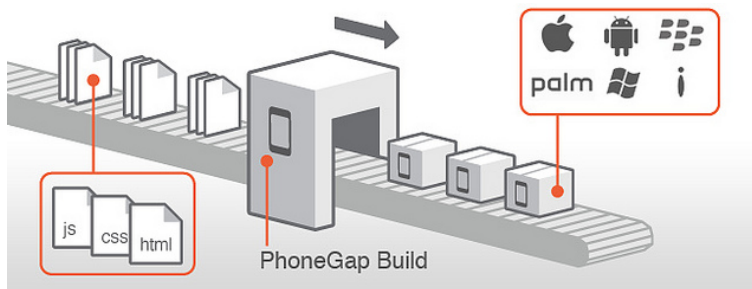


Figura 6.2: PhoneGap

PhoneGap [43] es un framework para el desarrollo de aplicaciones móviles producido por Nitobi, y comprado posteriormente por Adobe Systems. Principalmente, PhoneGap permite a los programadores desarrollar aplicaciones para dispositivos móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3. Las aplicaciones resultantes son híbridas, es decir que no son realmente aplicaciones nativas al dispositivo (ya que el renderizado es realizado mediante vistas web y no con interfaces gráficas específicas a cada sistema), pero no se tratan tampoco de aplicaciones web (teniendo en cuenta que son aplicaciones que son empaquetadas para poder ser desplegadas en el dispositivo incluso trabajando con el API del sistema nativo).

PhoneGap maneja APIs que permiten tener acceso a elementos como el acelerómetro, cámara, contactos en el dispositivo, red, almacenamiento, notificaciones, etc. Además PhoneGap permite el desarrollo ya sea ejecutando las aplicaciones en un navegador web, sin tener que utilizar un simulador dedicado a esta tarea, además, da la posibilidad de soportar funciones sobre frameworks como Sencha Touch o JQuery Mobile.

Utilizar phonegap tiene las siguientes ventajas:

- La principal y más obvia ventaja de PhoneGapes que se puede usar el mismo código fuente para las plataformas soportadas por el framework. Aunque lo anterior no es una verdad absoluta

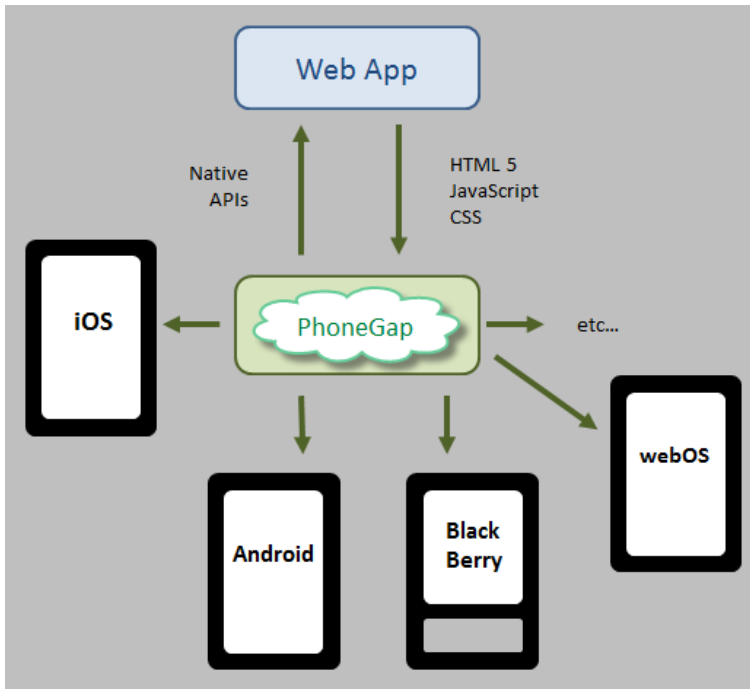


Figura 6.3: Funcionamiento de PhoneGap

dado que para algunas plataformas es necesario realizar unas pequeñas modificaciones en el código JavaScript y HTML de PhoneGap según características definidas por el fabricante de la plataforma nativa.

- En caso de que se deba implementar una aplicación simple que será ejecutada en diferentes plataformas, PhoneGap es una buena opción.
- Debido a que las funcionalidades disponibles para PhoneGap son más limitadas respecto a las disponibles en plataforma nativa, se requieren menos habilidades de implementación sobre aplicaciones móviles dentro del equipo de desarrollo.

Sin embargo Phonegap también tiene algunas desventajas:

- Dado que PhoneGap está diseñado para soportar diferentes plataformas, no contará con las características nuevas de cada plataforma tan pronto como estas se publiquen.
- Muchas de las características del núcleo de las plataformas podrán no estar disponibles vía PhoneGap.
- Para acceder a características no disponibles desde el framework (PhoneGap) se deben realizar llamados a implementaciones nativas (Por ejemplo, el acceso a SQLite solo es posible desde plataforma nativa). Esto tiene dos implicaciones; la primera es que la aplicación deja de ser multiplataforma debido a que se debe realizar una implementación nativa por funcionalidad no

disponible para cada plataforma. La segunda es que el medio definido por PhoneGap para intercambiar datos con la plataforma nativa es JSON, lo que incrementa de forma considerable el análisis de este tipo de estructuras deteriorando sustancialmente el desempeño de la aplicación.

- Debido a para implementar aplicaciones en PhoneGap se utiliza es JavaScript, html y css, existen potenciales problemas de compatibilidad con navegadores.

Para compilar aplicaciones web con PhoneGap, hay dos alternativas: Compilarlo localmente o compilarlo en la nube. La compilación local, que es la que se ha utilizado en este proyecto, consiste en descargarse los archivos de PhoneGap necesarios e incorporarlos al proyecto Android(o de la arquitectura utilizada en ese momento). La compilación en la nube se realiza a través de la herramienta PhoneGap Build.

PhoneGap Build requiere registrarse en la pagina de proyecto, y es gratuita para desarrolladores independientes, permitiendoles utilizar sus servicios para una sola aplicación privada, y todas las aplicaciones publicas que se desee.

Diseño responsivo

El diseño web adaptativo o adaptable (en inglés, Responsive Web Design) es una técnica de diseño y desarrollo web que mediante el uso de estructuras e imágenes fluidas, así como de media-queries en la hoja de estilo CSS, consigue adaptar el sitio web al entorno del usuario.

La idea y el propósito del diseño web adaptativo fue previamente discutida y descrita por el consorcio W3C en julio de 2008 en su recomendación "Mobile Web Best Practices" bajo el subtítulo "One Web".

Dicha recomendación, aunque específica para dispositivos móviles, puntualiza que está hecha en el contexto de "One Web", y que por lo tanto engloba no solo la experiencia de navegación en dispositivos móviles sino también en dispositivos de mayor resolución de pantalla como dispositivos de sobremesa.

6.4. Idea del juego

A continuación, se muestra un diseño preliminar de las pantallas de la aplicación. En la figura 6.4 se observa la idea inicial de la secuencia que tendrá el juego.

Splash Screen

La secuencia se inicia con una Splash Screen, que es la pantalla de carga de la aplicación. Es una pantalla opcional que se utilizará si a la aplicación le lleva mucho tiempo en iniciarse.

Menú principal

Esta es la pantalla principal de la aplicación. Desde ella se pueden acceder a los distintos mini-juegos propuestos. Así mismo se puede acceder a la configuración de la aplicación.

Minijuegos

En este apartado se engloban las pantallas correspondientes a los 8 minijuegos propuestos por el cliente.

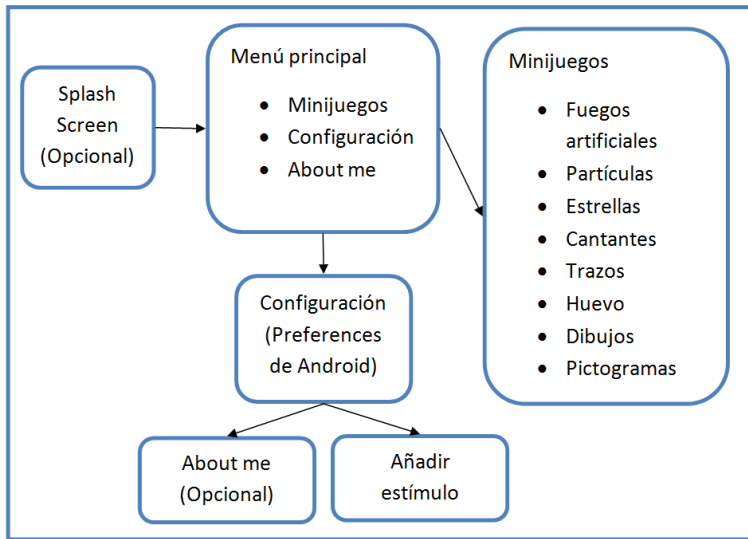


Figura 6.4: Idea del juego

Configuración

Esta es la pantalla donde se configura el sistema a la medida del usuario docente. Permite modificar parámetros diversos, así como de acceder a la pantalla de añadir estímulo.

About me

Esta pantalla es opcional, y contendría información del autor, como su nombre, correo electrónico o número de versión de la aplicación.

Añadir estímulo

Desde la configuración de la aplicación, se puede llegar a esta pantalla. En ella se permitirá al usuario añadir un estímulo a un determinado minijuego.

6.5. Canvas frente a Flash

Hasta la fecha la mayoría de los gráficos y animaciones interactivas de la web se realizan con Adobe Flash, a partir de ahora está tomando fuerza la opción Canvas/HTML5, en el futuro inmediato parece que seguirá usándose Flash hasta que Canvas sea capaz de realizar todo lo que Flash puede hacer.

Canvas es un elemento estándar de HTML5 y cualquier navegador debe ser capaz de manejarlo, mientras que Flash es una tecnología de una empresa que requiere instalar en el navegador un complemento (plug-in).

Hasta la fecha la mayoría de los navegadores tienen instalado el complemento de Flash mientras que todavía hay muchos navegadores instalados que no soportan Canvas. Sin embargo esto va a

cambiar rápidamente puesto que los navegadores nuevos que se van instalando no llevan Flash y sí soportan Canvas.

Desde el punto de vista del creador de gráficos es más fácil y rápido trabajar con Flash que con Canvas ya que Flash dispone de un entorno de diseño gráfico avanzado (Adobe Flash CS6) mientras que crear gráficos Canvas con JavaScript es más arduo. Esto también está cambiando con la aparición de frameworks para Canvas y entornos más amigables al estilo de Cookies.

Por otro lado Canvas es libre y abierto por lo que no hay que pagar licencia, mientras que Flash es propiedad de la empresa Adobe.

Respecto a lo que se puede hacer con cada sistema, hoy por hoy, los juegos complejos siguen haciéndose, mayoritariamente, con Flash.

Si tenemos que utilizar un gráfico a varias resoluciones, la tecnología vectorial de Flash es más eficiente que los mapas de bits de Canvas. En cambio, para gran parte de gráficos para la web, los gráficos Canvas se cargan más rápidamente que los de Flash.

Flash requiere más recursos para ejecutarse por lo que en dispositivos móviles Canvas lleva ventaja.

Otra alternativa para crear gráficos para la web es SVG que es un sistema vectorial, reconocido por W3C y soportado por los navegadores web modernos sin necesidad de instalar plug-in.

Por todo esto, se decide utilizar Canvas en lugar de Flash.

6.6. Diseño de la aplicación

Phonegap

Se ha decidido utilizar PhoneGap para el desarrollo de la aplicación. La razón fundamental de su uso consiste, en que phonegap permite que aplicaciones desarrolladas mediante elementos web tales como HTML5, CSS y JavaScript, sean portadas fácilmente a otras plataformas nativas tales como Android, iOS, o Windows Phone.

La figura 6.5 nos muestra la arquitectura de PhoneGap. En ella se observa como PhoneGap sirve de puente entre nuestra aplicación y los recursos nativos del sistema.

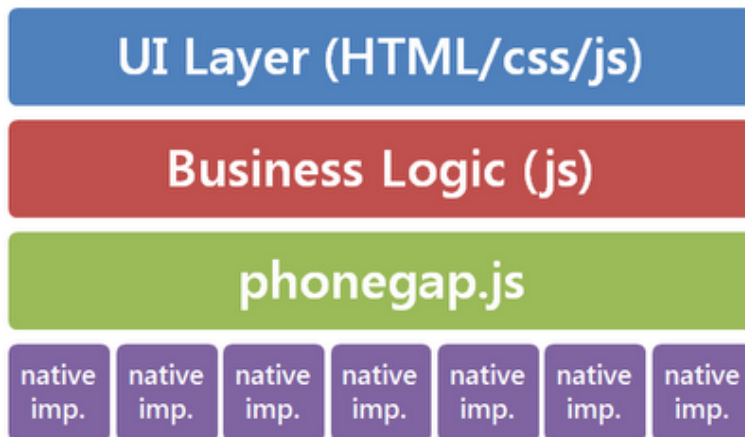


Figura 6.5: Arquitectura de PhoneGap

Una vez está claro que se va a utilizar una aplicación web, nos centramos en su diseño. Para diseñar la aplicación web, utilizaremos la estructura que se muestra en la figura 6.6.

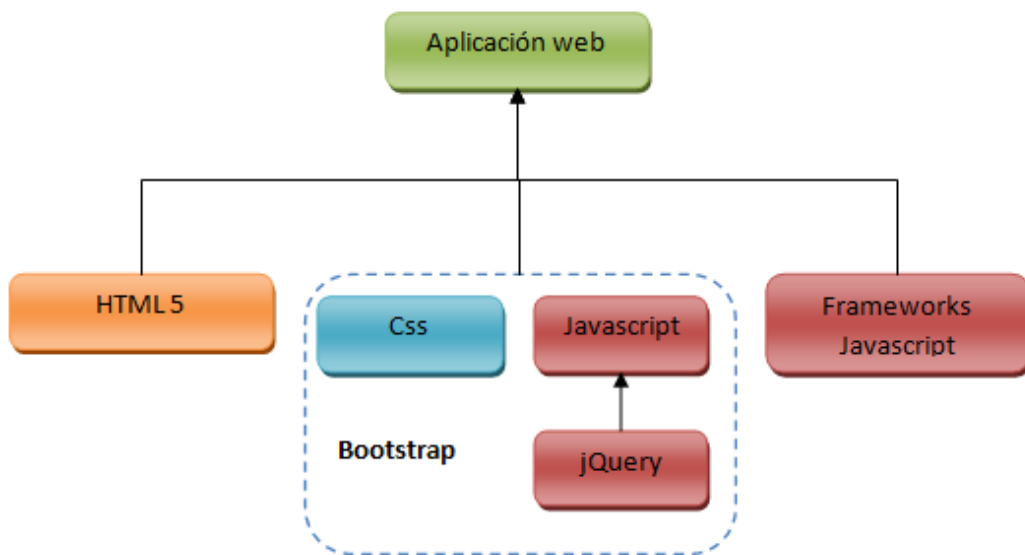


Figura 6.6: Estructura de la aplicación web

La aplicación web se compone de los siguientes elementos:

- **HTML5:** Última versión del lenguaje HTML. Se detalla la tecnología utilizada por HTML5 mas profundamente en la sección 3.2
- **Bootstrap:** Framework desarrollado por Twitter para crear interfaces y diseños web responsivos basados en HTML5 y CSS3. Se detalla con mas detalle los elementos que componen Bootstrap en la sección 3.9
- **Frameworks JavaScript:** Adicionalmente a lo ya expuesto, cada minijuego de la aplicación utilizará los frameworks JavaScript que sean necesarios para implementar la funcionalidad requerida.

6.6.1. Definir estilo

Un estilo es una guía o referencia estándar en el que se establece las directrices comunes para el formateo de textos e imágenes, el uso de colores y fuentes, las variaciones de logotipo que se pueden utilizar, etc. El objetivo es crear una consistencia a lo largo y ancho del proyecto así como servir de referencia a futuros desarrolladores.

Como hemos dicho anteriormente, hemos decidido usar el framework Bootstrap, ya que nos permite efectuar todo tipo de diseños muy fácilmente.

Colores

Se ha decidido utilizar colores vivos en las pantallas de la aplicación, ya que es uno de los requisitos de interfaz de usuario.

También se ha decidido que las ventanas de la aplicación tengan un pequeño reborde de un tono mas claro o mas oscuro que el del fondo de la ventana. En la tabla 6.1 pueden observarse los colores propuestos para la aplicación.

| Color de fondo | | Color del borde | |
|-------------------|-----------------|-------------------|---------------|
| Valor hexadecimal | Color | Valor hexadecimal | Color |
| F63A59 | Rojo anaranjado | F90D0D | Rojo |
| A4A4A4 | Gris | 6E6E6E | Gris Oscuro |
| 7FEB95 | Verde | 2E8640 | Verde Oscuro |
| DCE131 | Lima | F4FA58 | Amarillo |
| E3CEF6 | Lila | D7ADFC | Morado |
| F5BCA9 | Salmón | CF917D | Tostado |
| A9F5F2 | Azul Celeste | 0B8581 | Verde Botella |
| 58ACFA | Azul Claro | 0040FF | Azul |
| F781F3 | Rosa Chicle | F144EB | Rosa |

Tabla 6.1: Colores de fondo

Fuentes

Según la elicitación de requisitos, la letra debería ser "Escolar2".

Cuadrícula

Bootstrap proporciona un diseño de cuadrícula basado en filas y columnas donde se colocan los contenidos. La rejilla de Bootstrap funciona de la siguiente manera:

- Las filas se utilizan para agrupar horizontalmente a varias columnas.
- El contenido siempre se coloca dentro de las columnas, ya que las filas sólo deberían contener como hijos elementos de tipo columna.
- Las columnas de la rejilla definen su anchura especificando cuántas de las 12 columnas de la fila ocupan.

Esto tiene como consecuencia, que hay que modificar el diseño visual de algunas pantallas respecto al prototipo de baja definición.

El aspecto visual del menú principal no variará mucho del propuesto en la evaluación del prototipo no funcional, ya que la distribución gustó a los usuarios. En la figura 6.7 se puede observar un diagrama esquemático de la disposición de los distintos elementos que componen el menú principal.

El aspecto de visual de los minijuegos "Pictogramas", "Cantantes" y "Dibujos", varía respecto al prototipo funcional.

En la figura 6.8 se puede observar un diagrama esquemático de la disposición de los distintos elementos que componen estos minijuegos.

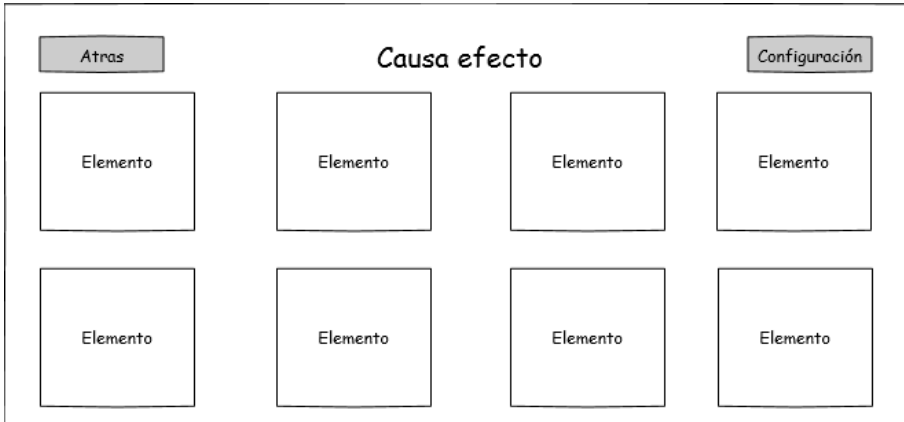


Figura 6.7: Diseño estructural de los elementos del menú principal

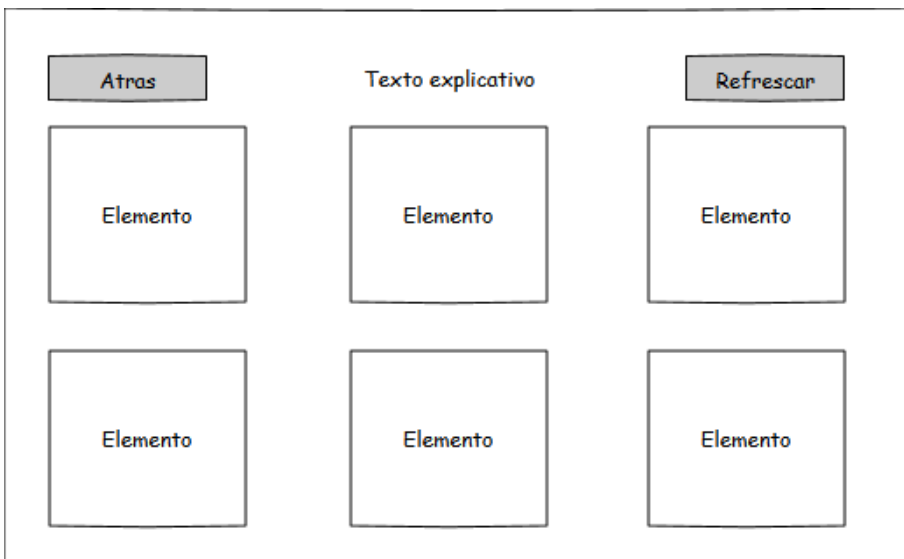


Figura 6.8: Diseño estructural de los elementos de los minijuegos "Pictogramas", "Cantantes" y "Dibujos"

El resto de pantallas de los minijuegos, tienen una rejilla consistente en una única columna con un único elemento(figura 6.9).

Imágenes

Las imágenes utilizadas en este proyecto provienen de diversas fuentes. Son las siguientes:

- Para el minijuego **canciones**, el cliente ha proporcionado un archivo word con una relacion de canciones infantiles e imagenes asociadas a estas canciones.
- Para el minijuego **cantantes**, se decide utilizar imágenes del rostro de los cantantes ya que al ser un rostro conocido son fácilmente identificables por los usuarios.

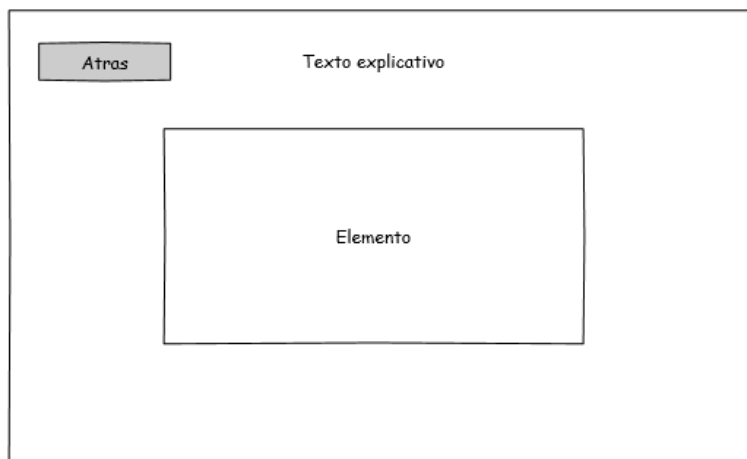


Figura 6.9: Diseño estructura del resto de ventanas de la aplicación

- En el minijuego **dibujos**, se decide utilizar capturas de pantalla en las que pueda leerse el nombre del episodio en cuestión. El motivo de hacerlo es diferenciar dos episodios diferentes de la misma serie.
- Los pictogramas se obtienen del Portal Aragonés de la Comunicación Aumentativa y Alternativa (ARASAAC) [14].

Se ha decidido que las imágenes se muestren en formato miniatura de imagen utilizando para ello el elemento de Bootstrap **thumbnail**.

Otros elementos gráficos

Se ha decidido usar los iconos (Glyphicons) proporcionados por Bootstrap para añadir un componente visual a los botones de la aplicación. Bootstrap incluye 180 iconos creados mediante una fuente especial llamada Glyphicon Halfings. Aunque esta fuente normalmente no es gratuita, su creador permite utilizar estos iconos gratuitamente dentro de Bootstrap 3.

6.6.2. Diseño de contenidos

Minijuego dibujos

Para el contenido del minijuego "*dibujos animados*" se ha recurrido a la página web de programación infantil de Radio Televisión Española¹. El motivo de elegir esa fuente de contenidos consiste en que el canal conocido como "clan" es un canal de televisión conocido por ser apto para el público infantil, con lo que el contenido que allí se muestra es válido para nuestra aplicación.

También se ha contado con los sobrinos del alumno, dos niños de cinco y tres años de edad sin deficiencia física o psíquica alguna. El objetivo de contar con estos niños, consiste en conocer sus gustos y opiniones sobre el material audiovisual presente en la página de RTVE para así, determinar mejor los contenidos más adecuados para los usuarios de la aplicación. El análisis de usuarios de la aplicación puede encontrarse en la sección 4.1 de la presente memoria.

¹<http://www.rtve.es/infantil/>



Figura 6.10: Página web de programación infantil de Radio Televisión Española

Los dibujos que se han decidido mostrar en el prototipo funcional son los siguientes:

- Chuggington
- DinoTren
- La abeja Maya
- PepaPig
- Pocoyo
- Wow! Wow! Wubbzy!
- Suckers

Minijuego cantantes

Para el contenido del minijuego *cantantes*, se han decidido mostrar los siguientes cantantes:

- Adele

- Amy Winehouse
- David Guetta
- Jennifer Lopez
- Justin Bieber
- Psy

Minijuego pictogramas

Para el minijuego de los *pictogramas*, se ha decidido usar la relación de canciones infantiles y pictogramas proporcionados por el propio centro Obregón.

Estos pictogramas son, entre otros los siguientes:

- El cocherito leré
- Tengo una vaca lechera
- El patio de mi casa
- Debajo de un botón
- En el coche de papá
- Tengo una muñeca vestida de azul

6.6.3. Diseño de persistencia

En esta sección se detalla el análisis de la persistencia de la aplicación.

En la figura 6.11 se muestra un esquema sobre la estructura jerárquica que tienen los datos que deben almacenarse de nuestra aplicación.

Se ha decidido que la persistencia de la aplicación se realice mediante archivos JSON. Detallo mas extensamente los archivos JSON en la sección 3.8

Configuración del minijuego Pictogramas

Los elementos que definen un pictograma son los siguientes:

- **Identificador:** Un numero entero correlativo que identifique de manera única cada pictograma.
- **Nombre:** Cadena de caracteres que identifique el pictograma.
- **Ruta de la imagen:** Ruta URL relativa o absoluta con la dirección de la imagen relacionada con el pictograma.
- **Ruta de la música:** Ruta URL relativa o absoluta con la dirección de la música relacionada con el pictograma.
- **Numero de pictogramas:** Valor entero que indique el numero de pictogramas que se desea mostrar en pantalla.

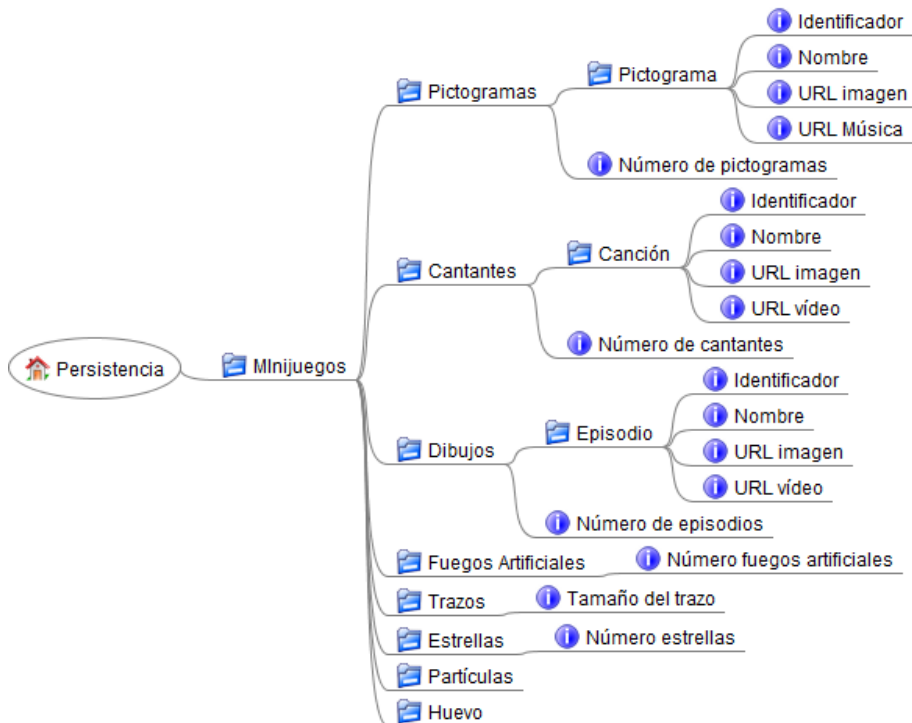


Figura 6.11: Mapa mental de la persistencia de la aplicación

Configuración del minijuego Cantantes

- **Identificador:** Un numero entero correlativo que identifique de manera única cada vídeo musical.
- **Nombre:** Cadena de caracteres que identifique el vídeo musical.
- **Ruta de la imagen:** Ruta URL relativa o absoluta con la dirección de la imagen relacionada con el vídeo musical.
- **Ruta del vídeo:** Ruta URL relativa o absoluta con la dirección del vídeo relacionada con el vídeo musical.
- **Numero de cantantes:** Valor entero que indique el numero de cantantes que se desea mostrar en pantalla.

Configuración del minijuego Dibujos

- **Identificador:** Un numero entero correlativo que identifique de manera única cada clip de vídeo.
- **Nombre:** Cadena de caracteres que identifique el clip de vídeo.
- **Ruta de la imagen:** Ruta URL relativa o absoluta con la dirección de la imagen relacionada con el clip de vídeo.

- **Ruta del vídeo:** Ruta URL relativa o absoluta con la dirección del vídeo relacionada con el clip de vídeo.
- **Numero de dibujos:** Valor entero que indique el numero de dibujos que se desea mostrar en pantalla.

Configuración del minijuego Fuegos artificiales

- **Numero de Fuegos artificiales:** Valor entero que indique el numero de fuegos artificiales que se desea mostrar en pantalla.

Configuración del minijuego trazos

- **Tamaño del trazo:** Valor entero que indique el tamaño del trazo que se va a mostrar por pantalla medido en píxeles.

Configuración del minijuego estrellas

- **Numero de estrellas:** Valor entero que indique el numero de estrellas que se desea mostrar en pantalla.

DCU: Implementación y Pruebas

En este capítulo, tratamos la implementación y pruebas del proyecto. Comentamos los pasos que se han dado para desarrollar el código de la aplicación. También realizamos una serie de pruebas, para asegurarnos que el producto final cumple con los requisitos propuestos en el capítulo 4.

7.1. Preparación del proyecto

Para crear un proyecto de las características dadas, en primer lugar, en el entorno eclipse seleccionaremos *Archivo - nuevo - Web project* tal y como aparece en la figura 7.1.

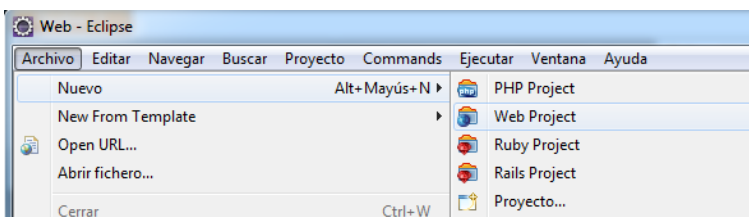


Figura 7.1: Creación del proyecto

Acto seguido, se selecciona la opción *Basic Web Template* (figura 7.2). Esta opción tan solo proporciona al proyecto un fichero *index.html* como base. Para finalizar se proporciona un nombre al proyecto, en este caso "*Causa-efecto*" y se finaliza el asistente.

7.2. Implementación del prototipo funcional

En esta sección se trata de las acciones que se han llevado a cabo para elaborar el prototipo funcional. Este prototipo está elaborado en HTML para ser mostrado en un navegador web. Es un paso previo para la implementación de la aplicación definitiva bajo Android.

7.2.1. Organización de las carpetas

En este apartado indicamos la organización de carpetas creadas utilizada en este proyecto.

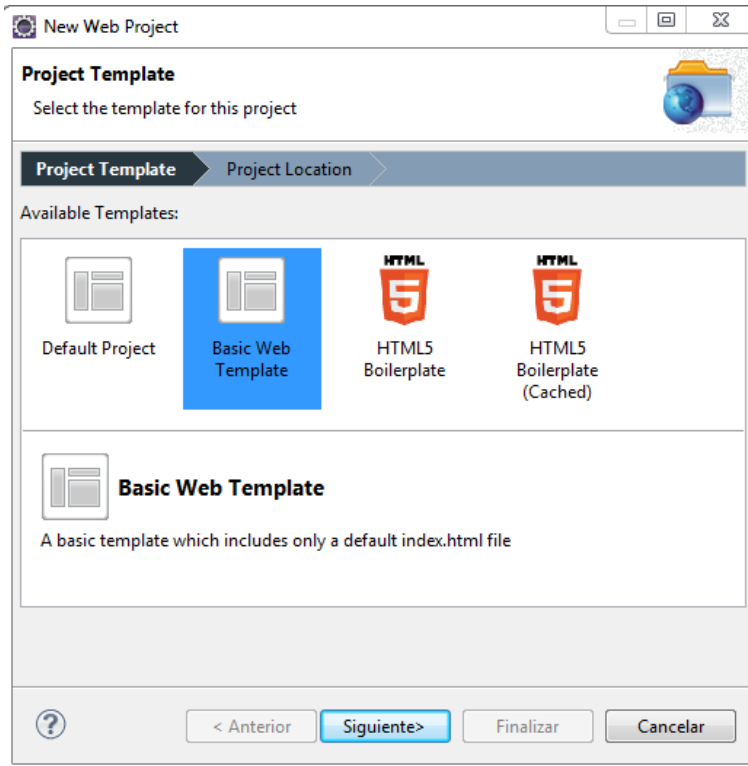


Figura 7.2: Creación del proyecto

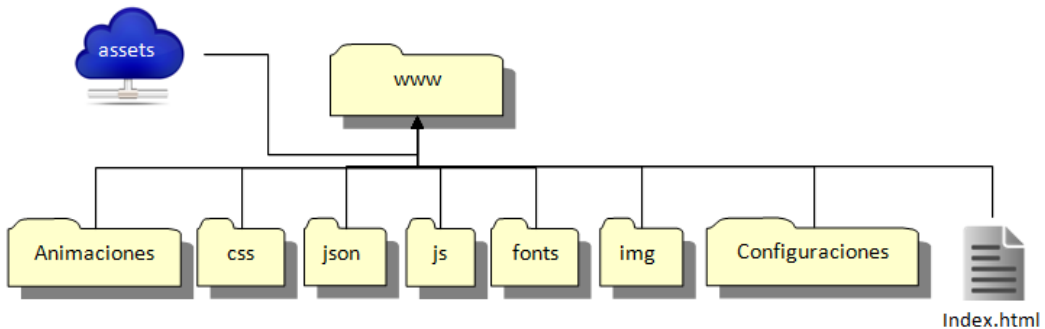


Figura 7.3: Organización carpetas HTML

- **assets:** Carpeta que contiene los recursos que necesitan los minijuegos. Es decir, contiene todas las imagenes, sonidos y videos que mostrarán los minijuegos.El contenido de la carpeta assets se muestra en la figura 7.4.
- **Animaciones:** Contiene las animaciones que se muestran en los minijuegos "Huevo" y "Fuegos artificiales".

- **css:** archivos css de bootstrap, las imágenes que se necesitan para algunos de los iconos que tienen ya definidos y mis hojas de estilos para terminar de definir el aspecto visual del proyecto.
- **json:** contiene los ficheros json con las referencias a los recursos (assets) utilizados en los minijuegos.
- **js:** contiene los ficheros javascript necesarios.
- **fonts:** Carpeta que contiene los ficheros con los iconos de Bootstrap así como la fuente "escolar2" .
- **img:** contiene las imágenes del proyecto.
- **configuraciones:** Carpeta con los ficheros de configuración del sistema.
- **index.html:** Fichero inicial que contiene el menú principal y lleva a el resto de ventanas de la aplicación. En este mismo nivel se encuentran el resto de ficheros html con cada uno de los minijuegos.

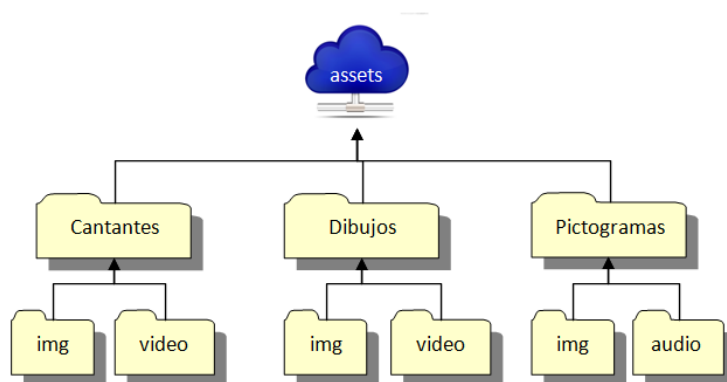


Figura 7.4: Organización de la carpeta assets

7.2.2. Bootstrap

Implementar Bootstrap en un proyecto es sumamente sencillo. No hay mas que descargar el paquete compilado de su pagina oficial ¹ y descomprimirlo en la carpeta del proyecto. En la figura 7.5 puede observarse la estructura de ficheros de Bootstrap.

7.2.3. Fuente

Uno de los requisitos del juego era usar un tipo especial de letra escolar. Para añadir un nuevo tipo de letra a la página utilizaremos CSS3. La regla @font-face permite crear una nuevo tipo de letra a partir de la url en donde se encuentra el fichero.

¹<http://getbootstrap.com/getting-started>



Figura 7.5: Estructura de Bootstrap

7.2.4. Iconos (glyphicons)

Los Iconos (glyphicons) vienen en el paquete de instalación de Bootstrap.

Por motivos de rendimiento, todos los iconos requieren de una clase CSS común para todos y de una clase CSS específica para cada uno. Para añadir un icono en cualquier elemento de la página, se utiliza código como el que se muestra a continuación.

```
<span class="glyphicon glyphicon-search"></span>
```

7.2.5. Thumbnails

Bootstrap 3 define varias clases CSS para decorar las imágenes de tus sitios web. Se decide utilizar la clase `thumbnail`, ya que esta clase muestra la imagen con un relleno blanco y un borde fino simulando el aspecto de las fotografías de las antiguas cámaras instantáneas. Añade además una breve animación para hacer que la imagen aparezca al cargar la página

Esta clase se utiliza con el código que se muestra a continuación.

```
<a href="#" class="thumbnail">  
    
</a>
```

7.2.6. Animaciones

Las animaciones de los minijuegos 'Huevo' y 'Fuegos artificiales' se han realizado utilizando el software **A5 HTML5 Animator**.

Este es un software para diseñar animaciones en HTML5 sin conocimientos de programación ni mucho menos Flash.

7.3. Integración con Phonegap

Implementar una aplicación Android con PhoneGap es muy sencillo. Los archivos necesarios para la implementación se descargan gratuitamente desde la página principal del proyecto [43]. De la carpeta descargada de PhoneGap se busca la se descomprime Hay que crear una aplicación Android en Eclipse, cuando está lista, dentro de la estructura del proyecto deben existir 2 carpetas (si no existen es necesario crearlas)

- libs/
- assets/www

De la carpeta descargada de PhoneGap buscamos la carpeta /lib/android y copiamos el archivo cordova.js en /assets/www. Luego copiamos el archivo cordova-2.8.0.jar en /libs (figura 7.6)

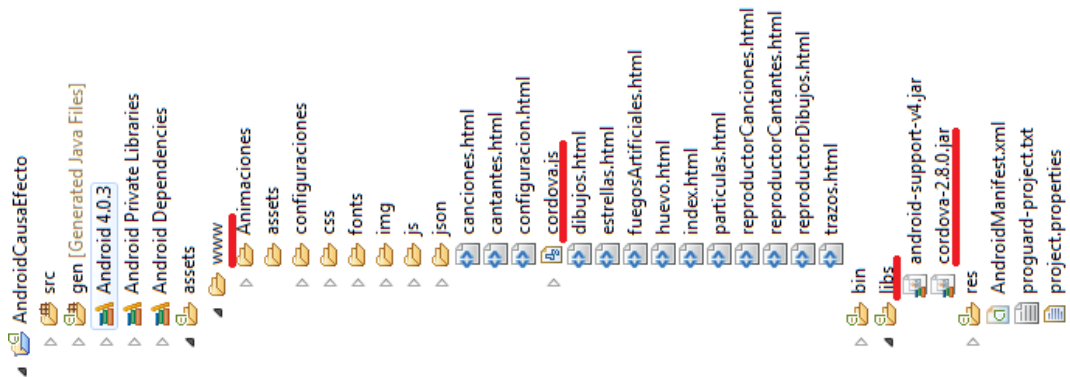


Figura 7.6: Instalación Phonegap paso 1

Luego en la carpeta /res creamos una carpeta llamada xml. En dicha carpeta copiamos el archivo config.xml que se encuentra en la carpeta de PhoneGap /lib/android/xml (figura 7.7)

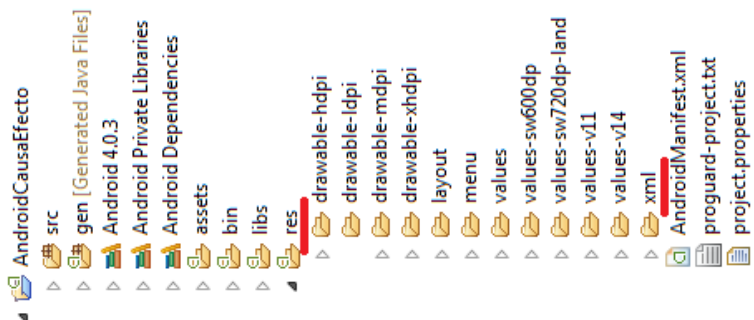


Figura 7.7: Instalación Phonegap paso 2

También es necesario modificar los archivos *MainActivity.java*, *AndroidManifest.xml* y se crea el archivo *index.html* dentro de la carpeta *www*.

Una vez hecho esto, ya estaría preparado el entorno de programación. A continuación copiaríamos los ficheros que creamos en el prototipo funcional.

7.4. Validación de la aplicación

Las pruebas son una parte muy significativa del proyecto de adecuación de las aplicaciones, no solo por su importancia en el logro de resultados correctos sino por el tiempo y recursos requeridos. El propósito del plan de pruebas es explicar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas. El objetivo del Plan de Pruebas es comparar los resultados obtenidos con los esperados para así:

- Identificar el origen de cada problema detectado para poder remitirlo a quien proceda, determinar la envergadura de las modificaciones y que acciones deben llevarse a cabo para resolverlo de forma satisfactoria.
- Indicar si el plan de pruebas debe volver a realizarse total o parcialmente, y si será necesario contemplar nuevos casos de prueba no considerados anteriormente.

La aplicación que vamos a probar es un prototipo de un juego "causa-efecto". El objetivo primordial de este Plan de Pruebas es asegurar el buen funcionamiento de la aplicación, así como adecuar los requisitos del cliente con las características técnicas del proyecto. Algunas de las pruebas están ligadas a otras, por lo que se realizarán en una secuencia de ejecución, otras pruebas son totalmente independientes, por lo que tendrían que realizarse en cualquier momento. Las pruebas deben realizarse con datos reales.

Las pruebas han de ser diseñadas con el fin de encontrar el máximo número de no conformidades con el mínimo esfuerzo y menor tiempo posible. Existen dos enfoques:

- **Caja negra:** Enfoque funcional. Con ellas se pretende demostrar que las funciones del software implementado son operativas, que la entrada es aceptada de una forma correcta y que como salida se obtiene la esperada.
- **Caja blanca:** Se basan en caminos lógicos del software. Se han de proponer casos en los que se ejerciten conjuntos específicos de condiciones.

El desarrollo de todas estas pruebas puede suponer cambios en la aplicación cuando como resultado de las mismas se encuentren aspectos mal implementados o que no concuerden con los requisitos pedidos por el cliente, es decir, cuando los resultados de las mismas no sean los esperados.

7.4.1. Metodología

El proceso de verificación y pruebas se ha realizado en el prototipo no funcional desplegado en una página web² propiedad del alumno.

Las características del equipo utilizado para verificar la aplicación son las siguientes:

- Sistema operativo Windows 7
- Navegador web Mozilla Firefox 30.0 y Google Chrome 35.0
- Conexión a Internet

²Url: <http://prignassur.factoriadigitalpremium.es/causaefecto/>

7.4.2. Pruebas de verificación

Las pruebas de verificación se centran en comprobar que el funcionamiento esperado de la aplicación concuerde con el funcionamiento real. En otras palabras, implica comprobar que el producto conforma la especificación inicial.

Las pruebas realizadas son las siguientes:

Menú principal

| | |
|------------------------------|---|
| Elemento de Prueba | CU-2: Seleccionar minijuego |
| Requisito relacionado | CU-2, RF1-Minijuegos, IU5 |
| Entradas esperadas | El educador accede al menú principal de la aplicación y selecciona uno de los minijuegos |
| Salidas esperadas | El sistema inicia el minijuego seleccionado por el educador |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | No |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada |

Tabla 7.1: Prueba de selección de minijuegos

| | |
|------------------------------|---|
| Elemento de Prueba | Carga de imágenes del menú principal |
| Requisito relacionado | Ninguno |
| Entradas esperadas | El educador accede al menú principal de la aplicación |
| Salidas esperadas | El sistema muestra correctamente todas las imágenes pertinentes |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | No |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada |

Tabla 7.2: Prueba de carga de imágenes del menú principal

Minijuego Estrellas

| | |
|------------------------------|--|
| Elemento de Prueba | Funcionamiento del minijuego Estrellas |
| Requisito relacionado | CU-3.3, RF8 |
| Entradas esperadas | El educador accede a la aplicación Estrellas |
| Salidas esperadas | El sistema muestra una pantalla con estrellas que se iluminan al tocarlas y que se pueden desplazar |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | Si |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada ■ El numero de estrellas no es configurable ■ El minijuego permite reiniciarse |

Tabla 7.3: Prueba de funcionamiento del minijuego Estrellas

Minijuego Huevo

| | |
|------------------------------|---|
| Elemento de Prueba | Funcionamiento del minijuego Huevo |
| Requisito relacionado | CU-3.6, RF6 |
| Entradas esperadas | El educador accede a la aplicación Huevo |
| Salidas esperadas | El sistema muestra un huevo, que al tocarlo, se resquebraja haciendo ruido y sale un pollito |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | No |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada ■ El minijuego permite reiniciarse |

Tabla 7.4: Prueba de funcionamiento del minijuego Huevo

Minijuego Dibujos

| | |
|------------------------------|---|
| Elemento de Prueba | Funcionamiento del minijuego Dibujos |
| Requisito relacionado | CU-3.7, RF4-Pantalla completa |
| Entradas esperadas | El educador accede a la aplicación Dibujos |
| Salidas esperadas | El sistema muestra varios dibujos animados. Al pulsar sobre ellos se reproduce el episodio |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | No |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada ■ Los vídeos pueden ser vistos a pantalla completa |

Tabla 7.5: Prueba de funcionamiento del minijuego Dibujos

| | |
|------------------------------|---|
| Elemento de Prueba | Carga de elementos multimedia del minijuego Dibujos |
| Requisito relacionado | CU-3.7, RF4-Pantalla completa |
| Entradas esperadas | El educador accede a la aplicación Dibujos |
| Salidas esperadas | El sistema carga correctamente todos los elementos multimedia(video e imagenes) del minijuego |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | No |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada |

Tabla 7.6: Prueba de carga del minijuego Dibujos

Minijuego Pictogramas

| | |
|------------------------------|---|
| Elemento de Prueba | Funcionamiento del minijuego Pictogramas |
| Requisito relacionado | CU-3.8, IU4-Pictogramas |
| Entradas esperadas | El educador accede a la aplicación Pictogramas |
| Salidas esperadas | El sistema muestra varios pictogramas asociados cada uno a una canción. Al pulsar se oye la canción |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | No |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada |

Tabla 7.7: Prueba de funcionamiento del minijuego Pictogramas

| | |
|------------------------------|---|
| Elemento de Prueba | Carga de elementos multimedia del minijuego Pictogramas |
| Requisito relacionado | CU-3.8, IU4-Pictogramas |
| Entradas esperadas | El educador accede a la aplicación Pictogramas |
| Salidas esperadas | El sistema carga correctamente todos los elementos multimedia(audio e imagenes) del minijuego |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | No |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada |

Tabla 7.8: Prueba de carga del minijuego Pictogramas

Minijuego Cantantes

| | |
|------------------------------|---|
| Elemento de Prueba | Funcionamiento del minijuego Cantantes |
| Requisito relacionado | CU-3.5, RF4-Pantalla completa |
| Entradas esperadas | El educador accede a la aplicación Cantantes |
| Salidas esperadas | El sistema muestra las imágenes de varios cantantes. Al pulsar sobre ellos se reproduce un video de su música |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | No |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada ■ Los vídeos pueden ser vistos a pantalla completa |

Tabla 7.9: Prueba de funcionamiento del minijuego Cantantes

| | |
|------------------------------|---|
| Elemento de Prueba | Carga de elementos multimedia del minijuego Cantantes |
| Requisito relacionado | CU-3.5, RF4-Pantalla completa |
| Entradas esperadas | El educador accede a la aplicación Cantantes |
| Salidas esperadas | El sistema carga correctamente todos los elementos multimedia(video e imágenes) del minijuego |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | No |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada |

Tabla 7.10: Prueba de carga del minijuego Cantantes

Minijuego Fuegos artificiales

| | |
|------------------------------|--|
| Elemento de Prueba | Funcionamiento del minijuego Fuegos artificiales |
| Requisito relacionado | CU-3.1, RF7, RF6 |
| Entradas esperadas | El educador accede a la aplicación Fuegos artificiales |
| Salidas esperadas | El sistema muestra varios fuegos artificiales y cohetes, que al tocarlos explotan |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | Si |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada ■ El numero de fuegos artificiales y cohetes no es configurable ■ El minijuego permite reiniciarse |

Tabla 7.11: Prueba de funcionamiento del minijuego Fuegos artificiales

Minijuego Trazos

| | |
|------------------------------|---|
| Elemento de Prueba | Funcionamiento del minijuego Trazos |
| Requisito relacionado | CU-3.4, RF6 |
| Entradas esperadas | El educador accede a la aplicación Trazos |
| Salidas esperadas | El sistema permite dibujar trazos de colores por la pantalla |
| Necesidades | <ul style="list-style-type: none"> ■ Navegador web conectado a Internet |
| Reporte de errores | No |
| Comentario | <ul style="list-style-type: none"> ■ El minijuego muestra la salida esperada ■ El minijuego permite reiniciarse |

Tabla 7.12: Prueba de funcionamiento del minijuego Trazos

Minijuego Partículas

| | |
|------------------------------|--|
| Elemento de Prueba | Funcionamiento del minijuego Partículas |
| Requisito relacionado | CU-3.2, RF6 |
| Entradas esperadas | El educador accede a la aplicación Partículas |
| Salidas esperadas | Al pulsar en la pantalla, el sistema dibuja partículas de colores que vuelan zumbando |
| Necesidades | <ul style="list-style-type: none"> ▪ Navegador web conectado a Internet |
| Reporte de errores | Si |
| Comentario | <ul style="list-style-type: none"> ▪ El minijuego no muestra la salida esperada ▪ El minijuego permite reiniciarse |

Tabla 7.13: Prueba de funcionamiento del minijuego Partículas

7.4.3. Resultados de las pruebas

Esta sección tiene como objetivo listar los resultados obtenidos al llevarse a cabo los casos de prueba especificados, todos ellos fueron registrados durante el proceso de prueba del sistema.

La tabla 7.14 que se muestra a continuación indica el estado de las pruebas realizadas:

- Éxito: Indica que se ha superado la prueba (el sistema actúa según lo esperado).
- Fallo: Indica que no se ha superado la prueba (el sistema no se comporta de forma adecuada).

7.5. Conclusiones

En este capítulo se ha tratado tanto de la implementación como de las pruebas realizadas a esta aplicación.

Se han comentado los pasos principales que se han tenido que dar para implementar esta aplicación.

La realización de pruebas a la aplicación es la manera de saber con seguridad que el sistema software que se va a entregar al cliente cumple con todos los requisitos que este exige.

Las pruebas realizadas han ayudado a detectar y corregir algunos pequeños problemas existentes en el prototipo funcional, sin embargo, algunos errores necesitan más trabajo y conocimiento para ser corregidos.

| Prueba | Respuesta | Estado |
|--|---|---------------|
| Seleccionar minijuego | El sistema inicia el minijuego seleccionado | Éxito |
| Carga imágenes menú principal | Se muestran todas las imágenes | Éxito |
| Funcionamiento estrellas | El sistema no permite configurar el numero de estrellas mostrados | Fallo |
| Funcionamiento Huevo | Muestra un huevo del que sale un pollito | Éxito |
| Funcionamiento Dibujos | Se muestran los dibujos animados deseados y se reproduce con exito los elementos multimedia asociados | Éxito |
| Carga elementos multimedia Dibujos | Se muestran todas las imágenes y vídeos | Éxito |
| Funcionamiento Pictogramas | Se muestran los pictogramas deseados y se reproduce con exito los elementos multimedia asociados | Éxito |
| Carga elementos multimedia Pictogramas | Se muestran todas las imágenes y sonidos | Éxito |
| Funcionamiento Cantantes | Se muestran los cantantes deseados y se reproduce con exito los elementos multimedia asociados | Éxito |
| Carga elementos multimedia Cantantes | Se muestran todas las imágenes y sonidos | Éxito |
| Funcionamiento Fuegos artificiales | El sistema no permite configurar el numero de fuegos artificiales mostrados | Fallo |
| Funcionamiento Trazos | Se muestran los trazos realizados por el usuario | Éxito |
| Funcionamiento Partículas | Las partículas no se generan desde la punta del dedo del usuario | Fallo |

Tabla 7.14: Resultados de las pruebas

Conclusiones y trabajos futuros

En este capítulo realizamos un resumen de los trabajos realizados a lo largo de este trabajo de fin de grado. Se analizan las principales aportaciones y se presentan las líneas de trabajo futuro. Se enumeran los objetivos cumplidos y se aporta una valoración personal del trabajo realizado.

8.1. Contribuciones

A lo largo de este trabajo de fin de grado, se han logrado los siguientes objetivos.

- El objetivo principal de este proyecto es realizar un juego de "causa-efecto" adaptado a los alumnos discapacitados del centro Obregón de Valladolid. Este objetivo no se ha alcanzado en su totalidad, sin embargo tratar con los alumnos discapacitados del centro Obregón ha despertado en este alumno la conciencia solidaria al comprobar de primera mano, las necesidades especiales que tienen los alumnos discapacitados del centro.
- Este proyecto ha servido para poner en práctica la metodología del diseño centrado en el usuario así como comprobar como esta metodología es capaz de mejorar la usabilidad y viabilidad de un producto software.
- Respecto a la parte técnica. Se ha aprendido a diseñar aplicaciones web con herramientas actuales tales como HTML 5, Bootstrap, jQuery, etc... Ha sido especialmente sorprendente descubrir la potencia de HTML5 en especial, de su elemento canvas.
- Ha sido muy interesante descubrir el framework Bootstrap, al que gracias a su flexibilidad, se ha convertido en el favorito de este alumno, a la hora de realizar otros trabajos basados en web.

8.2. Mejoras y trabajos futuros

Tras la realización de este proyecto, se plantean distintas líneas de trabajo:

- Finalizar el minijuego Partículas, para que realice su funcionalidad correctamente.
- Realizar una evaluación del prototipo funcional y pulir los fallos que detecte el cliente.
- Utilizar Phonegap para encapsular la aplicación y los recursos existentes al formato .apk de Android. Empaquetar los recursos(imágenes, sonidos y vídeos) tiene dos consecuencias. La primera consiste en que los recursos dejan de estar accesibles. La segunda consiste en que el

fichero .apk con la aplicación aumenta mucho en tamaño. Una solución consistiría en crear un asistente para ayudar a enlazar los recursos multimedia desde fuera de la aplicación. Ya sea ubicándolos en un servidor web como en la tarjeta SD de la tablet.

- Incluir mensajes de ayuda a lo largo de toda la aplicación.
- Implementar un sistema de persistencia donde almacenar las configuraciones de usuario a cada minijuego. Adicionalmente se podría incluir un sistema de gestión de perfiles de usuarios centralizado. De manera que todas las tablets del centro Obregón obtuvieran el perfil de un alumno del sistema de persistencia y no fuera necesario configurar cada tablet por separado.

Definición de términos y abreviaturas

Accesibilidad: Posibilidad de que un producto o sitio web pueda ser accedido y usado por el mayor número posible de personas, indiferentemente de las limitaciones propias del individuo o de las derivadas del contexto de uso

AOSP: Android Open Source Project. Proyecto dedicado al desarrollo y manejo de Android desde el punto de vista de código abierto, libre de todo lo referente a fabricantes y sus especificaciones.

AOT: Ahead Of Time. Compilación antes de tiempo. Acto de compilar un programa en código maquina, antes de la ejecución del programa, en lugar de compilar durante la ejecución.

API: (Application Program Interface). Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones.

Aplicación: Cada uno de los programas que una vez ejecutados, permiten trabajar con el ordenador.

CSS: Cascading Style Sheets. Hojas de Estilo en Cascada. Es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación.

DCU: Diseño centrado en el usuario.

DOM: Document Object Model. Estándar de objetos para representar documentos HTML y XML. A través de el, los programas pueden acceder y modificar el contenido, estructura y estilo.

Framework: Esquema para el desarrollo y/o la implementación de una aplicación.

HTML: HyperText Markup Language. Lenguaje de marcado de hipertexto.

JSON: JavaScript Object Notation. Notación de Objetos de JavaScript. Formato ligero de intercambio de datos.

LESS: Herramienta para la generación de archivos CSS.

PHP: Lenguaje de programación de código abierto especialmente adecuado para el desarrollo web.

Responsive web design: Diseño web adaptativo. Técnica de diseño y desarrollo web que mediante el uso de estructuras e imágenes fluidas, así como de media-queries en la hoja de estilo CSS, consigue adaptar el sitio web al entorno del usuario.

ROM: contiene los archivos y código necesario para que un dispositivo arranque y pueda ejecutar Android.

Stakeholder: Cualquier persona o entidad que es afectada por las actividades de una organización.

Usabilidad: Capacidad de un producto software de ser entendido, aprendido, usado y atractivo para el usuario, cuando es usado bajo unas condiciones específicas.

Widget: es una pequeña aplicación o programa que se presenta como una pequeña ventana o caja. En general, se usa para tener acceso rápido a programas o funciones usadas frecuentemente, como la calculadora, reloj o el calendario,

Contenido del CD-ROM

Todos los ficheros generados a lo largo del desarrollo de este proyecto han sido incluidos en el CD-ROM que acompaña esta memoria. He aquí una breve descripción del contenido de su árbol de directorios:

- Aplicación:
 - Aplicación web: Este directorio contiene el código fuente de la web.
 - assets: Contiene los elementos multimedia usados en esta aplicación.
- Documentación:
 - Memoria: La presente memoria, en formato PDF, y el archivo fuente creado con \LaTeX .

Bibliografía

- [1] 148APPS: «App Store Metrics», 2014.
<http://148apps.biz/app-store-metrics/>
- [2] ADSL.: «Manual básico de jQuery».
http://mundosica.github.io/tutorial_hispano_jQuery/sesion01/index.html
- [3] APPBRAIN: «Number of available Android applications», 2014.
<http://www.appbrain.com/stats/number-of-android-apps>
- [4] AULACLIC: «Introducción a Canvas. Gráficos y animaciones en HTML 5.», 2013.
http://www.aulacli.es/articulos/canvas_1.html
- [5] BLEIGH, MICHAEL y INTRIDEA, INC.: «Sketch.js», 2011.
<http://intridea.github.io/sketch.js/>
- [6] BOOTSTRAP: «Bootstrap», 2013.
<http://getbootstrap.com/>
- [7] BRACHET, PASCAL: «Texmaker : original LaTeX editor since 2003 by the creator of Kile», 2011.
<http://www.xmlmath.net/texmaker/>
- [8] CAMPAÑA, FERNANDO.: «El Modelo de Caja».
<http://www.scourdesign.com/articulos/tutoriales/css/css12.php>
- [9] CARRERO, ANGEL.: «10 Framework para desarrollar aplicaciones para dispositivos móviles basadas en HTML».
http://www.programacion.com/articulo/10_framework_para_desarrollar_aplicaciones_para_dispositivos_moviles_basadas_en_html_498
- [10] CINCO, HTML: «Las 15 APIs más populares de HTML5», 2012.
<http://www.htmlcinco.com/15-apis-de-html5/>
- [11] CORPORATION., MICROSOFT: «Definición de HTML», 2011.
<http://msdn.microsoft.com/es-es/ie/hh749020>
- [12] —: «Microsoft by the Numbers», 2014.
<http://www.microsoft.com/en-us/news/bythenumbers/index.html>

- [13] DAWEED: «Obtener parametros de la url con jQuery», 2012.
<http://da-software.blogspot.com.es/2012/02/obtener-parametros-de-la-url-con-jquery.html>
- [14] DE ARAGÓN, GOBIERNO: «ARASAAC», 2013.
<http://catedu.es/arasaac/index.php>
- [15] —: «¿Qué son los Sistemas Aumentativos y Alternativos de Comunicación (SAAC)?», 2013.
<http://arasaac.org/aac.php>
- [16] DE CASO PARRA, ASTOR: JavaScript Edición 2010 (Guia Práctica). Anaya, 2010.
- [17] DEVELOPERS, ANDROID: «Android, the world's most popular mobile platform», 2014.
<https://developer.android.com/intl/es/about/index.html>
- [18] —: «Plataform Versions», 2014.
<https://developer.android.com/about/dashboards/index.html>
- [19] DEVERIA, ALEXIS: «Can I use...», 2014.
<http://caniuse.com/>
- [20] EGUILUZ, JAVIER.: «Introducción a CSS».
<http://librosweb.es/css/>
- [21] FRAMINGHAM, MASS: «Android and iOS Continue to Dominate the Worldwide Smartphone Market with Android Shipments Just Shy of 800 Million in 2013, According to IDC». IDC, 2014.
<http://www.idc.com/getdoc.jsp?containerId=prUS24676414>
- [22] GAUCHAT, J.D.: El gran libro de HTML5, CSS3 y Javascript. EL GRAN LIBRO DE. Marcombo, 2012. ISBN 9788426717825.
<http://books.google.es/books?id=szDm1RzwzuUC>
- [23] GIRONES, JESUS TOMAS: El gran libro de android. Marcombo, 2012.
- [24] GOULD, JOHN D. y LEWIS, CLAYTON: «Designing for Usability: Key Principles and What Designers Think». Commun. ACM, 1985, **28(3)**, pp. 300–311. ISSN 0001-0782. doi: 10.1145/3166.3170.
<http://doi.acm.org/10.1145/3166.3170>
- [25] HASSAN-MONTERO, YUSEF y ORTEGA-SANTAMARÍA, SERGIO: Informe APEI sobre usabilidad. Asociación Profesional de Especialistas en Información, 2009.
<http://eprints.rclis.org/13253/>
- [26] HTML-COLOR CODES.INFO: «Códigos de colores HTML», 2011.
<http://html-color-codes.info/codigos-de-colores-hexadecimales/>
- [27] IAZZA, YUSEF HASSAN & FRANCISCO J. MARTÍN FERNÁNDEZ & GHZALA: «Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información». Hipertext.net, 2004.
http://www.upf.edu/hipertextnet/numero-2/disenho_web.html
- [28] INTERNATIONAL, ECMA: «ECMAScript Language Specification», 2011.
<http://www.ecma-international.org/publications/files/ECMA-ST/ECma-262.pdf>

-
- [29] LAWTON, SHAWN: Simplemente pregunta: Integración de la accesibilidad en el diseño. ET/Lawton, 2008.
<http://www.uiaccess.com/JustAsk/es/>
- [30] MACKAY, MARK: «El modelo 500 de Henry Dreyfuss», 2009.
<http://blog.duopixel.com/articulos/dreyfuss.html>
- [31] MONTERO, Y.H.; SANTAMARÍA, S.O.; FERNÁNDEZ, R.L. y ASOCIACIÓN PROFESIONAL DE ESPECIALISTAS EN INFORMACIÓN (GIJÓN, ESPAÑA): Informe APEI sobre usabilidad. Informe APEI. Asociación Profesional de Especialistas en Información, 2009. ISBN 9788469237823.
<http://www.nosolousabilidad.com/manual/index.htm>
- [32] NEEKS.COM: «JavaScript avanzado: eliminar un elemento por valor en un array», 2011.
<http://www.ngeeks.com/javascript-avanzado-eliminar-un-elemento-por-valor-en-un-arr>
- [33] —: «JavaScript: funciones básicas para arrays», 2011.
<http://www.ngeeks.com/javascript-funciones-arrays-length-concat-join-pop-push-shif>
- [34] NORMAN, DONALD A: «Design principles for human-computer interfaces». En: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, ACM, 1983.
- [35] PIÑAN, CARLOS: «Introducción al desarrollo de juegos en Android», 2011.
<http://www.slideshare.net/igdape/introduccion-al-desarrollo-de-juegos-en-android>
- [36] POSADA PRIETO, FERNANDO: «Diseño de Materiales multimedia. Web2.0». Instituto Superior de Formación y Recursos en Red para el Profesorado, 2008.
<http://www.ite.educacion.es/formacion/materiales/107/cd/imagen/imagen0105.html>
- [37] PREECE, JENNY: Interaction Design: Beyond Human-Computer Interaction. Wiley Academic, 2002.
- [38] ROWELL, ERIC: «Html5CanvasTutorials», 2013.
<http://www.html5canvastutorials.com>
- [39] SHMIDDTY: «Unable to clear canvas». <http://stackoverflow.com>, 2012.
<http://stackoverflow.com/questions/13805424/unable-to-clear-canvas>
- [40] SIEMER, ROBERT.: «Which Android runs which Linux kernel?». <https://android.stackexchange.com/questions/51651/which-android-runs-which-linux-kernel>
- [41] SOUNDBIBLE.COM: «SoundBible.com», 2013. Free Sound Clips, Sound Bites, and Sound Effects.
<http://soundbible.com/>
- [42] SPAIN, PHONEGAP.: «Acerca de».
<http://www.phonegapspain.com/acerca-de/>
- [43] SYSTEMS, ADOBE: «PhoneGap», 2013.
<http://phonegap.com/>
-

- [44] TARJUCCINO: «Introducción a JSON», 2014.
<http://tarjuccino.com/tutoriales/programacion-web/introduccion-a-json/>
- [45] TEAM, JABREF DEVELOPMENT: JabRef. JabRef Development Team, 2010.
<http://jabref.sourceforge.net>
- [46] THE JSON GROUP: «Introducción a JSON».
<http://www.json.org/json-es.html>
- [47] TWISTEDWAVE: «TwistedWave Online», 2012.
<https://twistedwave.com/online/>
- [48] VALDIVIESO, CLAUDIA: «Instalacion de Phonegap para Android».
<http://lavaldi.blogspot.com.es/2013/02/instalacion-de-phonegap-para-android.html>
- [49] VALLADOLID, ASPRONA: «Asprona Valladolid».
<http://www.asprona-valladolid.es/>
- [50] VILLA, RAÚL DE: «PhoneGap vs. Código Nativo», 2012.
<http://rauldevilla-movilidad.blogspot.com.es/2012/04/phonegap-vs-codigo-nativo.html>
- [51] VOS, TANJA: «Usabilidad en aplicaciones informáticas», 2008.
<http://www.iti.es/media/about/docs/tic/08/articulo2.pdf>