

RA 1 Teoría Parte 2 – Tecnologías de desarrollo para dispositivos móviles

Configuraciones. Tipos y características.

Perfiles. Características. Arquitectura y requerimientos. Diseño y Desarrollo

Ciclo de vida - Desarrollo Software.

Modificación de aplicaciones existentes.

Utilización del entorno de ejecución del administrador de aplicaciones.

Configuraciones. Tipos y características.

Las aplicaciones se pueden clasificar atendiendo a diversos criterios, entre ellos:

- **a. Por sus efectos psicosociales o psicopedagógicos:**
 - Aplicaciones capacitadoras: aquellas que permiten o incitan a buscar posibilidades nuevas o fomentar la creatividad.
 - Aplicaciones de dependencia: aquellas que impiden, limiten o determinen nuestros actos, capacidad de elección, creatividad, etc.
- **b. Por el tipo de contenido que ofrecen al usuario:**
 - De entretenimiento: donde se encuadran mayoritariamente las apps de juegos.
 - De relación social: dirigidas a la comunicación interpersonal
 - De producción o utilitarias: proporcionan instrumentos para la resolución de tareas específicas que requieren inmediatez y rapidez para solucionar problemas, en especial en el sector empresarial y comercial.
 - Educativas o informativas: diseñadas y desarrolladas como transmisoras de la información y el conocimiento donde se prioriza el acceso a los contenidos y a las herramientas de búsqueda mediante una interfaz de navegación lo más sencillo y fácil posible.
 - Creativas: ofrecen herramientas que potencien la creatividad literaria, musical (y sonora), fotográfica o video-gráfica.
 - Publicitarias: con fines comerciales la gran mayoría son de distribución gratuita.

Configuraciones. Tipos y características.

- **c. Por las condiciones de distribución:** Pueden clasificarse como gratuitas, de pago y *freemium*, las cuales permiten su descarga inicial gratuita para un uso limitado y básico, posibilitando posteriormente el acceso a funcionalidades más avanzadas previo pago.
- **d. Por la edad de destino de los usuarios del contenido:** El App Store establece una clasificación del contenido por tramos de edades de “4+, 9+, 12+ y 17+”, que limita el acceso a la descarga de dicha aplicación.
- **e. Por el tipo de diseño y desarrollo:** Como ya se ha especificado en apartados anteriores su diseño y desarrollo permite diferenciar entre aplicaciones.
 - Genéricas: prácticamente todo el diseño y programación de lenguaje es compatible con la mayoría de los dispositivos.
 - Híbridas: determinados componentes de la programación son comunes para todos los teléfonos inteligentes y otro porcentaje es específico, dependiendo del sistema operativo.
 - Nativas: su programación en su totalidad es específica para cada Market de distribución.

Perfiles. Características. Arquitectura y requerimientos. Diseño y Desarrollo

El proceso de diseño y desarrollo de una aplicación, según CUELLO y VITTONI, se puede estructurar en cinco etapas secuenciadas en los siguientes apartados:

Conceptualización

- La aplicación parte de una idea que permita cubrir una necesidad o facilitar una actividad en el mundo real de un determinado sector de población en función de sus necesidades y problemas. La idea debe responder a las expectativas factibles y concretas, lo que implica la necesidad de realizar un análisis prospectivo de la viabilidad del concepto que se quiere crear.

Perfiles. Características. Arquitectura y requerimientos. Diseño y Desarrollo

Definición

- Determinada la posibilidad de acometer el proyecto, diseñador/es y desarrollador/es proceden a definir las funcionalidades de la aplicación en consonancia con el perfil de los usuarios y las especificaciones técnicas, con objeto establecer, los parámetros de acceso al hardware del dispositivo, si va a ser una aplicación específica para cada tienda (nativa) o híbrida,... El dimensionado de todo ello permitirá determinar el alcance del proyecto, su duración, coste económico y complejidad del diseño y programación de la aplicación.

Perfiles. Características. Arquitectura y requerimientos. Diseño y Desarrollo

Diseño

- En esta etapa se materializan los aspectos de la etapa anterior (especificaciones, funcionalidades, etc.). Para ello se realiza, en primer lugar, un diseño esquemático sin gráficos (“wireframe”) que será testeado por un grupo usuarios. Superada esta prueba inicial el diseño definitivo será entregado al desarrollador en archivos y pantallas separadas para que añada el código de programación. Los sistemas operativos permiten interactuar con el usuario presentando en la pantalla los elementos necesarios para ello de forma distinta, lo cual debe tenerse en consideración por parte de los diseñadores de acuerdo con las siguientes premisas:[6](#)

Perfiles. Características. Arquitectura y requerimientos. Diseño y Desarrollo

- **Desarrollo**

El programador, en función del tipo de aplicación diseñada se encarga de dar vida a los diseños y crear la estructura sobre la cual se apoyará el funcionamiento de la aplicación, creando el código funcional mediante un lenguaje de programación.

- **Publicación**

Una aplicación se publica tras un período de pruebas, su correcto y estable funcionamiento, sin errores de usabilidad y diseño, y cumpliendo las políticas y requerimientos de las tiendas. Durante toda la vida útil de la aplicación, es necesario un seguimiento analítico, estadístico y de comentarios de usuarios. El fin es evaluar el comportamiento y funcionalidad de la app, detectar y corregir errores, y realizar mejoras o actualizaciones.

Ciclo de vida – Desarrollo Software

- El **Proceso para el desarrollo de software**, también denominado **ciclo de vida del desarrollo de software**, es una estructura aplicada al desarrollo de un producto de software. Hay varios modelos a seguir para el establecimiento de un proceso para el desarrollo de software, cada uno de los cuales describe un enfoque diferente para diferentes actividades que tienen lugar durante el proceso.
- Algunos autores consideran un modelo de ciclo de vida un término más general que un determinado proceso para el desarrollo de software. Por ejemplo, hay varios procesos de desarrollo de software específicos que se ajustan a un modelo de ciclo de vida de espiral.

Ciclo de vida de una aplicación

- **Planificación**

La importante tarea a la hora de crear un producto de software es obtener los requisitos o el análisis de los requisitos. Los clientes suelen tener una idea más bien abstracta del resultado final, pero no sobre las funciones que debería cumplir el software.

- **Implementación, pruebas y documentación**

La implementación es parte del proceso en el que los ingenieros de software programan el código para el proyecto de trabajo que está en relación de las demandas del software, en esta etapa se realizan las pruebas de caja blanca y caja negra.

Las pruebas de software son parte esencial del proceso de desarrollo del software. Esta parte del proceso tiene la función de detectar los errores de software lo antes posible.

La documentación del diseño interno del software con el objetivo de facilitar su mejora y su mantenimiento se realiza a lo largo del proyecto. Esto puede incluir la documentación de un API, tanto interior como exterior. Prácticamente es como una receta de cocina

Ciclo de vida de una aplicación

- **Despliegue y mantenimiento**

El despliegue comienza cuando el código ha sido suficientemente probado, ha sido aprobado para su liberación y ha sido distribuido en el entorno de producción.

Entrenamiento y soporte para el software es de suma importancia y algo que muchos desarrolladores de software descuidan. Los usuarios, por naturaleza, se oponen al cambio porque conlleva una cierta inseguridad, es por ello que es fundamental instruir de forma adecuada a los futuros usuarios del software.

El mantenimiento o mejora de un software con problemas recientemente desplegado, puede requerir más tiempo que el desarrollo inicial del software. Es posible que haya que incorporar código que no se ajusta al diseño original con el objetivo de solucionar un problema o ampliar la funcionalidad para un cliente. Si los costes de mantenimiento son muy elevados puede que sea oportuno rediseñar el sistema para poder contener los costes de mantenimiento.

Modificación de aplicaciones existentes.

- En nuestro caso particular en el cual usaremos el Motor Gráfico Unity nos encontraremos que muchas aplicaciones están desarrolladas con versiones inferiores a la nuestra, o en algún caso concreto pudiera ser superior.
- Cuando vas a lanzar un proyecto desde una versión que no es con la que se he desarrollado, Unity te advierte de dicha situación y te da la posibilidad de migrar el desarrollo a la nueva versión sin hacer nada adicional.
- Por lo general la integridad del proyecto no va a afectar en nada entre diferentes versiones. A no ser que la diferencia entre versiones sea muy grande, que puede dar fallos de configuración por software obsoleto por ejemplo.

Utilización del entorno de ejecución del administrador de aplicaciones.

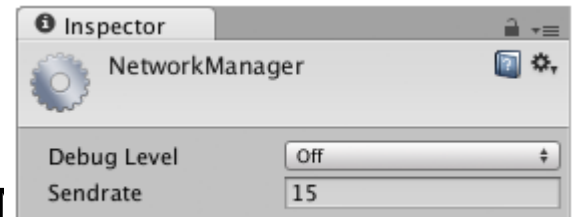
- Los Settings Managers de Unity están disponibles desde el menú Edit > Project Settings. Los ajustes afectan aspectos generales de la funcionalidad de Unity, tal como las Gráficas, Física y los detalles del reproductor publicado.
- Los tipos de administración se exponen a continuación.

Utilización del entorno de ejecución del administrador de aplicaciones.

- **Audio Manager** (Administrador de Audio) - El Audio Manager le permite ajustar el volumen máximo de todos los sonidos reproduciéndose en la escena. Para verlo escoja Edit > Project Settings > Audio.
- **Ajustes del Editor** - La Configuración del editor le permite elegir opciones que afectan la forma en que trabaja con el editor.
- **Administrador de Input** (Input Manager) - El Input Manager es dónde usted define todos los diferentes ejes de input(entrada) y acciones de juego para su proyecto.

Utilización del entorno de ejecución del administrador de aplicaciones.

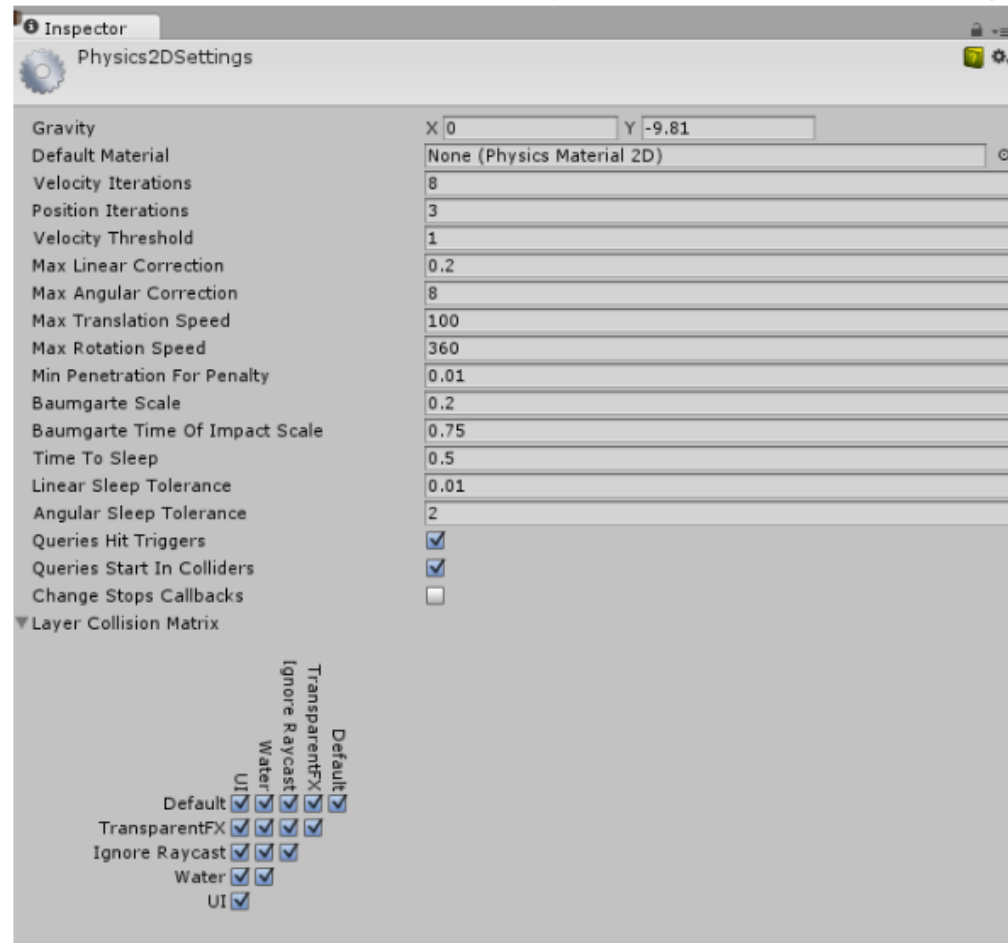
- **Administrador de Red (Network Manager)** - El Network Manager contiene dos propiedades muy importantes para la creación de juegos multijugador en red, Debug Level y SendRate.



- **Administrador de Física (Physics Manager)** –El Physics Manager permite a usted proporcionar ajustes globales para las físicas de 3D (menú: Edit > Project Settings > Physics). También hay un administrador correspondiente para físicas de 2D.
- **Administrador de Física en 2D(Physics 2D Manager)** - Usted puede acceder el Physics 2D Manager desde el menú seleccionando Edit->Project Settings->Physics2D.

Utilización del entorno de ejecución del administrador de aplicaciones.

- Administrador de Física en 2D(Physics 2D Manager)



Bibliografía

- https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil
- <https://docs.unity3d.com/es/530/Manual/comp-ManagerGroup.html>
- [https://es.wikipedia.org/wiki/Proceso para el desarrollo de software](https://es.wikipedia.org/wiki/Proceso_para_el_desarrollo_de_softwa_re)