



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

ELABORACIÓN DE GUÍA DE DESARROLLO PARA APLICACIONES INTERACTIVAS EN UN ENTORNO SECOND SCREEN (DCADHA-13)

Autor: Anush Virginia Euredjian Chiappe

Director: Diego Álvarez



Índice

Índice de tablas	6
Índice de figuras	7
Introducción.....	8
Propósito.....	9
Definición.....	11
Contexto	11
Alcance.....	12
Resumen de la Metodología.....	15
Acercamiento utilizado para el desarrollo	15
Características de XP.....	17
Organización del equipo del proyecto.....	19
Herramientas de desarrollo y colaboración utilizadas	21
Persistencia de la información	22
Control de cambios	24
Control de los datos.....	22
Planificación	25
Diagrama de procesos	27
Casos de uso.....	28
Planificación temporal	37
Tecnologías Contempladas	39
Opciones contempladas.....	39
Apple: iOS y Airplay.....	39
Google TV.....	40
LG SmartTV	40
Samsung SmartTV y AllShare Framework	41
Desarrollo	42
Versión 0.1 – Nombre <i>Abigail</i> (Alpha).....	42
Descripción	42
Problemas conocidos	42
Cambios	42
A destacar en esta versión.....	42
Versión 0.1.1 (Alpha)	44
Descripción	44
Novedades respecto a versión anterior.....	44
Problemas solventados.....	44
Problemas conocidos	44
Cambios	44
A destacar en esta versión.....	44
Versión 0.1.2 (Alpha)	45
Descripción	45
Novedades respecto a versión anterior.....	45
Problemas solventados.....	45
Problemas conocidos	45
Cambios	45
A destacar en esta versión.....	46
Versión 0.1.3 (Alpha)	46
Descripción	46

Novedades respecto a versión anterior	46
Problemas solventados	46
Problemas conocidos	46
Cambios	46
A destacar en esta versión	46
Versión 0.1.4 (Alpha)	47
Descripción	47
Novedades respecto a versión anterior	47
Problemas solventados	47
Problemas conocidos	48
Cambios	48
A destacar en esta versión	48
Versión 0.1.5 (Alpha)	49
Descripción	49
Novedades respecto a versión anterior	49
Problemas solventados	49
Problemas conocidos	49
Cambios	49
A destacar en esta versión	49
Versión 0.1.6 (Alpha)	50
Descripción	50
Novedades respecto a versión anterior	50
Problemas solventados	50
Problemas conocidos	50
Cambios	50
A destacar en esta versión	51
Versión 0.2 – Nombre Bibi (Alpha)	53
Descripción	53
Novedades respecto a versión anterior	53
Problemas solventados	53
Problemas conocidos	53
A destacar en esta versión	53
Junta de revisión 03/07/2013	53
Versión 0.2.1	53
Descripción	53
Novedades respecto a versión anterior	53
Problemas solventados	54
Problemas conocidos	54
Cambios	54
A destacar en esta versión	54
Versión 0.2.2	54
Descripción	54
Novedades respecto a versión anterior	54
Problemas solventados	54
Problemas conocidos	54
Cambios	54
A destacar en esta versión	54
Junta de revisión 08/07/2013	55
Problemas y/o mejoras detectadas:	55
Versión 0.2.3 / 0.2.4	55
Descripción	55
Novedades respecto a versión anterior	55
Problemas solventados	55
Problemas conocidos	55
Cambios	55

A destacar en esta versión	55
Versión 0.3 – Nombre Cook (Alpha)	57
Descripción	57
Novedades respecto a versión anterior	57
Problemas solventados	57
Problemas conocidos	57
Cambios	57
A destacar en esta versión	57
Junta de revisión 15/07/2013	57
Problemas y/o mejoras detectadas	57
Versión 0.3.1	58
Descripción	58
Novedades respecto a versión anterior	58
Problemas solventados	58
Problemas conocidos	58
Cambios	58
A destacar en esta versión	58
Junta de revisión 18/07/2013	59
Problemas y/o mejoras detectadas	59
Versión 0.3.2	59
Descripción	59
Novedades respecto a versión anterior	59
Problemas solventados	59
Problemas conocidos	59
Cambios	59
A destacar en esta versión	59
Junta de revisión 22/07/2013	60
Problemas y/o mejoras detectadas	60
Versión 0.3.3	60
Descripción	60
Novedades respecto a versión anterior	60
Problemas solventados	60
Problemas conocidos	60
Cambios	60
A destacar en esta versión	60
Junta de revisión 24/07/2013	61
Problemas y/o mejoras detectadas	61
Versión 0.4 - Effy (Beta)	62
Descripción	62
Novedades respecto a versión anterior	62
Problemas solventados	62
Problemas conocidos	62
Cambios	62
Resultados y posibles mejoras	63
Resultado final	63
Pantalla de Splash	63
Pantalla de Login	64
Pantalla de registro.....	65
Pantalla de información.....	66
Pantalla Principal	67
Pantalla de información de proyecto	69
Pantalla de proyecto.....	69
Otras mejoras	70
Panel de administración.....	71

Conclusiones	73
Referencias Bibliográficas	75
Enlaces de interés	76
Anexo I: Instalación del SDK.....	78
Requerimientos para instalación	78
Preparación para la instalación	79
Instrucciones de instalación	79
Entorno Linux	79
Entorno Windows	83
Creación de nuestro directorio de trabajo	87
Anexo II: Instalación de la aplicación en la TV	88
Preparando el servidor	88
Preparando la SmartTV	90
Anexo III: Apigee vs. API propia	92
Node.js.....	93

Índice de tablas

Tabla 1 - Comparación de Metodologías.....	15
Tabla 2 - Caso de Uso 1	29
Tabla 3 - Caso de Uso 2	30
Tabla 4 - Caso de Uso 3.1	31
Tabla 5 - Caso de Uso 3.2	32
Tabla 6 - Caso de Uso 4.1	33
Tabla 7 - Caso de Uso 4.2	34
Tabla 8 - Caso de Uso 5.....	35
Tabla 9 - Progreso temporal	37

Índice de figuras

Ilustración 1 - Arquitectura aplicación.....	14
Ilustración 2 - Prácticas XP	18
Ilustración 3 - Estructura eXtreme Programming.....	19
Ilustración 4 - Estructura UCD	20
Ilustración 5 - Estructura de la API	23
Ilustración 6 - Panel de control Github.....	24
Ilustración 7 - Diagrama BPMN	27
Ilustración 8 - Diagrama Casos de Uso	36
Ilustración 9 - Estructura SDK Samsung.....	41
Ilustración 10 - Pantalla de Splash	63
Ilustración 11 - Pantalla de login	64
Ilustración 12 - Pantalla de login	64
Ilustración 13 - Pantalla de registro	65
Ilustración 14 - Pantalla de información	66
Ilustración 15 - Pantalla principal	67
Ilustración 16 - Pantalla principal	67
Ilustración 17 - Pantalla principal	68
Ilustración 18 - Pantalla de información de proyecto.....	69
Ilustración 19 - Pantalla de proyecto	69
Ilustración 20 - Panel de Administración	71
Ilustración 21 - Panel de Administración	71
Ilustración 22 - Panel de Administración	72
Ilustración 23 - Anexo I. Instalación Unix	80
Ilustración 24 - Anexo I. Instalación Unix	80
Ilustración 25 - Anexo I. Instalación Unix	81
Ilustración 26 - Anexo I. Instalación Unix	81
Ilustración 27 - Anexo I. Instalación Unix	81
Ilustración 28 - Anexo I. Instalación Unix	82
Ilustración 29 - Anexo I. Instalación Windows	83
Ilustración 30 - Anexo I. Instalación Windows	84
Ilustración 31 - Anexo I. Instalación Windows	84
Ilustración 32 - Anexo I. Instalación Windows	85
Ilustración 33 - Anexo I. Instalación Windows	86
Ilustración 34 - Anexo II. Configuración Servidor	89
Ilustración 35 - Anexo II. Configuración Servidor	89
Ilustración 36 - Anexo II. Configuración SmartTV	90
Ilustración 37 - Anexo II. Configuración SmartTV	90
Ilustración 38 - Anexo II. Configuración SmartTV	90
Ilustración 39 - Anexo II. Configuración SmartTV	91
Ilustración 40 - Anexo II. Configuración SmartTV	91

Introducción

Este proyecto final de carrera surge con la idea de involucrarse en el desarrollo de una aplicación para *Samsung SmartTV*, haciendo uso de su SDK. Tras utilizar las comúnmente conocidas como "televisiones inteligentes" o *SmartTV's* nos dimos cuenta que había una carencia notable de aplicaciones, así como una falta de calidad en las mismas.

Si dedicamos unos minutos a navegar por la tienda de Samsung *SmartHub*, nos percataremos de que el contenido no es tan solo escaso, sino también poco variado. El contenido disponible mayormente abarca aplicaciones oficiales de periódicos, canales de televisión, redes sociales y canales de contenido multimedia. Es poco común encontrar aplicaciones que difieran en naturaleza de las anteriormente mencionadas, y aquellas que si varían dejan bastante que desear en el ámbito estético e interactivo.

Las oficiales no por ser oficiales mejoran mucho en contenido, concretamente, si investigamos un poco por las aplicaciones disponibles, vemos que a día de hoy¹ apenas habrá unas 40 en el mercado español, muchas de las cuales están ya preinstaladas en el televisor, aumentando hasta el centenar en mercados extranjeros como puede ser el británico. El hecho de que por ahora solo se puedan ofertar aplicaciones gratuitas en el mercado español, puede suponer sin duda un factor influyente sobre esta realidad.

Pese a todo esto consideramos que el mercado de las *SmartTV* es un mercado en auge y relativamente nuevo. Muchas de sus características cambian la forma en la que estamos acostumbrados a "ver" la televisión. Samsung nos permite instalar aplicaciones en nuestros televisores, hace una realidad del control por voz y gestos, facilita la conexión a internet (con todo lo que ello conlleva), entre otras muchas posibilidades. Es por todo esto por lo que nos parece interesante explotar el potencial de estos dispositivos de la marca Samsung.

Asimismo, a través de la escuela tuvimos la oportunidad de conocer la iniciativa *analizo.info*, dándonos la posibilidad de desarrollar un proyecto real y con un enfoque específico. La principal característica de esta plataforma es la de intentar convertir al consumidor en analista de la información que consume, todo ello enfocado al cambio, con un fuerte factor social y bajo los principios de OpenData y OpenSource.

No solo se pretende crear una herramienta para dicha iniciativa, pese a ser este sin duda el principal objetivo. Esta guía pretende involucrarse en un proyecto real además de servir de orientación para futuros desarrolladores que pretendan abordar un proyecto de similares características. A lo largo de este documento aspiraremos a desglosar, documentar y explicar a mayor nivel de detalle posible las decisiones tomadas y la naturaleza de las mismas con el fin de que sirva de guía y apoyo, no solo a los miembros de *analizo.info*, sino también a cualquiera de la comunidad de desarrolladores, a fin de sustentar y fomentar un entorno de colaboración.

¹ <http://www.samsung.com/es/tv-apps> - Último acceso 26/08/2013.

Propósito

El objetivo una vez acabado este proyecto es tener una aplicación que permita analizar contenidos audiovisuales durante su emisión por parte de las cadenas de TV. Para ello, junto a mi compañero que está realizando de manera paralela el proyecto “*DESARROLLO DE APLICACIÓN INTERACTIVA PARA TABLET Y SMARTTV EN UN ENTORNO "SECOND SCREEN - (DCADHA-14)*”, hemos decidido coordinar los enfoques utilizados en ambos proyectos para crear una aplicación efectiva y funcional.

Desde esa perspectiva, pretendemos crear una aplicación nativa en el televisor permitiendo definir y manejar proyectos dentro de la misma herramienta, de modo que se pueda analizar de forma sencilla y rápida la temática y contenido del mismo bajo ciertos parámetros facilitando a *analizo.info* la integración con sus proyectos.

En este documento se encuentra la definición del proyecto, así como nuestras metodologías de trabajo y el proceso alcanzado para obtener la solución final. Con ello se pretende, no solo mostrar el código que hace posible el funcionamiento de la misma, si no ir un paso más allá y documentar cuales son las limitaciones del entorno y porqué se han tomado ciertas decisiones u obviado otras aparentemente válidas.

Con este proyecto aspiramos a desarrollar una aplicación para plataformas *SmartTV*, que no sólo mejore visual y funcionalmente a las ya disponibles en el mercado, sino también con una finalidad social. El personal de la iniciativa *analizo.info* a efectos prácticos actuará como cualquier cliente, solicitando una solución software. Nuestra labor en todo momento será la de intentar ofrecer un producto eficaz, robusto y atractivo, explotando al máximo nuestras capacidades y el entorno, sin despreciar las limitaciones que se nos pueden presentar en una plataforma tan poco explorada como es el caso.

Empezaremos entrando en detalle sobre la forma de trabajo que emplearemos, analizando y justificando las distintas metodologías, herramientas y protocolos que vamos a utilizar en nuestro desarrollo. A fin de documentar en la medida de lo posible las decisiones tomadas en este proyecto, realizaremos un breve análisis de las distintas opciones barajadas. Este análisis se realizará a título informativo y en él explicaremos los motivos que nos llevaron a descartar su uso o implantación. Posteriormente, y a un nivel de detalle superior al de los anteriores apartados, se documentará el desarrollo y el resultado obtenido.

Algunas de estas decisiones nos vienen dadas por la plataforma. Por ejemplo, debido a que el *Software Development Kit (SDK)* de *Samsung SmartTV* hace uso de HTML y librerías Javascript, estos serán los lenguajes utilizados para el desarrollo de la aplicación.

El desarrollo se realizará mediante técnicas de *eXtremeProgramming (XP)*, la cual hace uso a menor escala de prototipos en sus tareas. Esta opción nos permite la posibilidad de obtener una retroalimentación con el usuario de forma rápida y

continua, lo cual hará que haya que documentar varias revisiones e iteraciones a lo largo de la presente memoria.

Como ya hemos indicado, estas elecciones serán debidamente justificadas en posteriores apartados.

Definición

Contexto

El principal propósito y aspiración de *analizo.info* es conseguir y promover la participación activa de la ciudadanía en procesos de análisis de información, es una iniciativa de emprendimiento social creada para facilitar la participación activa de la ciudadanía en procesos de análisis de información.

El portal *www.analizo.info* enfatiza el papel que desempeña la información en los procesos de transformación social. La iniciativa se configura como una comunidad online dirigida tanto a personas interesadas en poner en marcha procesos de análisis de información abiertos, sin ánimo de lucro y que tengan como objetivo el cambio social; como a todas aquellas personas que quieran transformar su entorno, apoyando los procesos de análisis iniciados por otras personas. La plataforma *analizo.info* requiere de la participación ciudadana y la colaboración para sacar adelante procesos de análisis que persigan difundir información a la opinión pública y generar resultados en conjunto con otros ciudadanos.

Los proyectos incluidos en la comunidad por parte de las personas promotoras se dan a conocer a usuarios y usuarias para que elijan si participan en el proyecto como analistas. Los promotores y promotoras proporcionan a los analistas los objetivos perseguidos, la justificación sobre las potencialidades de transformación social, la información a analizar, las herramientas para realizar el proceso de análisis y la metodología que garantice la consecución de resultados objetivos.

Cualquier usuario de la Red podrá acceder a las noticias e información de los proyectos contenidos en *analizo.info*, pero solo podrán participar como analistas los usuarios y usuarias registradas.

La información analizada junto a los resultados obtenidos estarán en abierto a disposición de la Sociedad para que puedan ser usados, analizados desde otras perspectivas o difundidos. *Analizo.info* impulsa la información en abierto como parte del movimiento **Open Data y Open Research**.

Al igual que otras plataformas movimientos y proyectos globales que han inspirado la iniciativa, tales como *datakind.org*, *unglobalpulse.org*, *change.org*, *access-info.org*, *Fundación Civio*, *The Open Knowledge Foundation*, *socialnest.org*, *goteo.org* o *verkami.com*, desde *analizo.info* se trabaja para conseguir un mundo en que la gente tenga poder para generar cambios en lo cotidiano, acceder al análisis colectivo de información y divulgar de forma abierta los resultados. Hay una voluntad de mejorar la sociedad y lograr el aumento de los bienes y recursos comunes.

Analizo.info se suma a estas iniciativas como la primera comunidad española formada por ciudadanos y ciudadanas para participar en procesos de análisis de información que persigan la mejora de la sociedad. Desde que se puso en marcha a finales de 2012, cuenta ya con 221 miembros.

Alcance

Con el fin de facilitar la labor de *analizo.info*, así como promover y facilitar el uso de nuevas tecnologías, decidimos crear una herramienta de análisis directamente en la fuente. De esta forma los analistas acceden a la información en el momento, y no dependen de pesados sistemas de BigData, que restan dinamismo y en ocasiones ralentizan el acceso a dicha información. Así, en vez de recopilar grandes cantidades de datos, en este caso programas de televisión, para analizarlos posteriormente, llevamos el análisis de los datos directamente a la fuente, y es el usuario analista de la aplicación el que en tiempo real consigue discernir los datos que realmente son importantes.

Actualmente, una de las fuentes de información más empleadas es la televisión. Pese a haber sido relegada en los últimos años a un segundo plano debido a internet², sigue siendo una de las mayores redes informativas. A fin de poder crear una herramienta de análisis integrada con este medio, decimos proponer este proyecto, que nos permitiría desarrollar una aplicación para *SmartTV* que además pudiese ser utilizada y aprovechada por usuarios reales.

Nuestra aspiración es la de crear una aplicación nativa en el televisor cuyas estadísticas y datos se compartan con *analizo.info*. La idea es poder gestionar y trabajar con aquellos proyectos de *analizo.info* que abarquen el medio televisivo. Permitiendo definir y manejar varios proyectos dentro de la misma herramienta, restringidos a un canal y franja horaria predefinidos, de modo que se pueda analizar de forma sencilla y rápida la temática y contenido del mismo bajo ciertos parámetros

A fin de facilitar este análisis, dentro de la aplicación cada proyecto tendrá ciertas opciones entre las que los analistas elegir. El contenido a analizar dentro de cada proyecto variará en naturaleza, por lo que para facilitar el uso de la aplicación y limitar el espectro del análisis se le ofrecerá al usuario un número limitado de opciones a elegir a través del mando a distancia. Si una vez dentro de la aplicación, el usuario se encuentra analizando un proyecto y observa en la emisión algún comportamiento que encaja dentro de las categorías previamente establecidas por *analizo.info*, bastará con que envíe una votación seleccionando el botón oportuno. A modo de ejemplo, un usuario podrá acceder a un proyecto en el que se analice el tiempo dedicado en los informativos a hablar de política internacional. Dicho proyecto contendrá ciertas opciones de entre las que elegir, por ejemplo "menos de 5 minutos", "entre 5 y 10 minutos", "entre 10 y 15 minutos" o "más de 15 minutos". Una vez el usuario decida que posee la suficiente información para emitir su opinión tan solo bastará con que seleccione la opción más acorde a sus observaciones.

Por supuesto, además de lo anteriormente mencionado, la funcionalidad básica que deberá tener conlleva el control de identidad, control horario y canal sintonizado, ayuda e instrucciones de uso, así como obviamente el envío de votaciones e información recogido desde la aplicación nativa a la web.

² Penetración superior a un 70% en España (2) - Último acceso 26/08/2013

El programa ha de ejecutarse en un primer plano, sin embargo debe poder verse la imagen del canal sintonizado sin problemas. Uno de los retos a los que nos enfrentaremos será crear un equilibrio visualmente atractivo, al forzar que la aplicación y la imagen compartan pantalla en un mismo espacio.

Esto tiene un propósito, la idea es que el usuario pueda visualizar el contenido con normalidad, y en un momento dado a raíz de un evento o noticia, tenga la posibilidad de registrar un suceso que encaje dentro de las opciones disponibles en el análisis que se está llevando a cabo. Si cada vez que se tuviera que elegir una opción el usuario se viera forzado a abrir la aplicación desde el menú de aplicaciones de su *SmartTV* esto no sería factible, pues el momento habría pasado.

Para lograr esto de forma efectiva la restricción de canal y franja horaria es importante, acotando el momento y situación en la que un usuario envía la opción marcada, de este modo aseguraremos una rigurosidad y fiabilidad documental. Hay que tener en cuenta que cada proyecto tendrá sus parámetros y restricciones, cada proyecto deberá tener un canal asociado, por lo que al seleccionarlo la propia aplicación deberá sintonizar el canal específico al proyecto seleccionado. Del mismo modo, si un proyecto no se selecciona en el horario para el que fue predefinido no se podrá acceder a él, ni seleccionar las opciones de un proyecto si el tiempo del mismo se ha visto superado.

Al finalizar este proyecto y habiendo respetado las políticas descritas en líneas superiores, obtendremos un herramienta de análisis eficaz, que además tenga una finalidad y vaya a tener un uso y explotación mas allá de la entrega de esta memoria. Desde sus inicios se concibe como un método para facilitar el análisis, tanto a *analizo.info* como a sus usuarios.

Esperamos que al desarrollar una herramienta para un medio tan popular como es la televisión, la comunidad de analistas de *analizo.info* se vea incrementada. Sin duda un mayor número de participantes no solo facilitará el análisis, sino también afianzará la información obtenida al poder ofrecer un mayor abanico de resultados.

En resumen la aplicación consistiría en lo siguiente:

- Desde *analizo.info* se introducen los proyectos y contenido a analizar.
- La aplicación en la *SmartTV* obtiene estos datos desde internet y los gestiona para el usuario.
- El usuario, mientras ve la televisión, se encarga de analizar el contenido que le interesa a *analizo.info* en tiempo real y la aplicación lo guarda en internet.
- Estos datos son recogidos por *analizo.info* y para la realización de estadísticas y estudios.

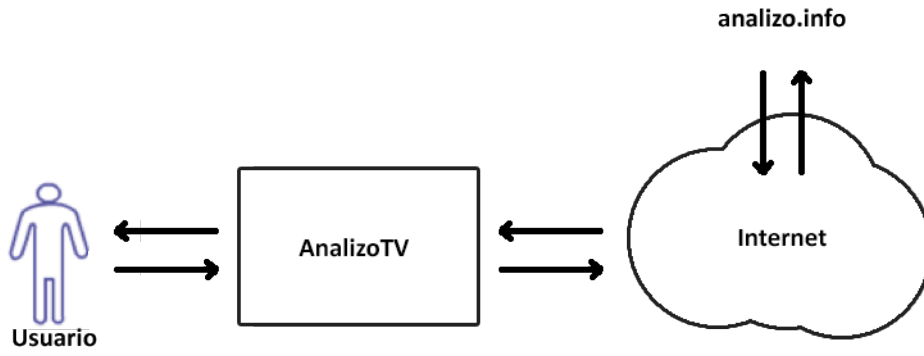


Ilustración 1 - Arquitectura aplicación

Resumen de la Metodología

Acercamiento utilizado para el desarrollo

El desarrollo de software no es una labor sencilla. Prueba de ello es que existen numerosas propuestas metodológicas con distintas dimensiones y etapas en el proceso de desarrollo.

Por una parte tenemos aquellas propuestas conocidas como "tradicionales" (Beck, K, 99), que se centran especialmente en el control del proceso, definiendo rigurosamente las actividades, artefactos, herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado muchos problemas. Una forma de evitar estos contratiempos es incluir en la planificación del desarrollo más actividades, más diagramas y más restricciones, basándose en los posibles puntos débiles. No obstante, se obtendría un proceso de desarrollo más complejo y tedioso, que no solo prolongaría sino que dependiendo de la naturaleza del proyecto, puede incluso limitar el éxito del proyecto.

Otras aproximaciones se centran en factores muy diferentes, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software, la gran mayoría con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad.

Lo anteriormente resumido, se puede ver de una forma mucho más gráfica en la siguiente tabla, donde se comparan las principales diferencias entre las metodologías anteriormente mencionadas:

Metodologías ágiles	Metodologías tradicionales
Se basan en heurísticas provenientes de prácticas de producción de código	Se basan en normas provenientes de estándares seguidos por el entorno de desarrollo
Preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente por el equipo	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso muy controlado, numerosas normas
Contrato flexible e incluso inexistente	Contrato prefijado
El cliente es parte del desarrollo	Cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10)	Grupos grandes
Pocos artefactos	Más artefactos
Menor énfasis en la arquitectura del software	La arquitectura del software es esencial

Fuente: Canós, J *et al*, 2005. Metodologías Ágiles.

Tabla 1 - Comparación de Metodologías

Debido a la naturaleza del proyecto decidimos investigar e intentar abordar este nuevo proyecto con una metodología ágil. También nos vimos influenciados por experiencias anteriores con metodologías de desarrollo más tradicionales, donde dedicamos mucho tiempo al análisis y especificación del proyecto pero notamos carencias durante el proceso de programación, convirtiéndolo en una tarea pesada y muy rígida, donde cualquier error detectado podía suponer un serio problema para el éxito del proyecto.

A lo largo de nuestra investigación dimos con varias metodologías. como pueden ser: *SCRUM*, *Crystal Methodologies*, *Dynamic Systems Development Method (DSDM)*, *Adaptive Software Development (ASD)*, *Feature-Driven Development (FDD)*, *Lean Development (LD)* y *eXtremme Programming (XP)* (Letelier, P. y Sánchez, E., 2003)

Las que mejor se adaptan a nuestro proyecto fueron SCRUM y XP, ya que de las anteriormente mencionada estas dos suelen tener una duración menor y funcionar bien con grupos reducidos de desarrolladores. Para elegir entre estas dos metodologías tuvimos en cuenta diferentes factores, ambas son metodologías ágiles iterativas, lo que cambia son las formas de trabajar y las exigencias de cada una.

SCRUM está más estructurado en comparación a XP. Tiene responsabilidades más señaladas aunque en ambas se implica al cliente muy directamente, este factor no se ve afectado. La principal diferencia es en el modo de trabajo de los desarrolladores. Mientras que con SCRUM un grupo se dedica en tiempo completo a un desarrollo, XP realiza su desarrollo a tiempo parcial y por parejas. Básicamente, SCRUM prioriza el proyecto y XP un entorno de trabajo cómodo para los desarrolladores. En cuanto al diseño, en XP se concibe al principio para posteriormente programarse, iterando a medida que se crean versiones. Al final se aplica una refactorización para mejorar el diseño final. En cambio, SCRUM se realiza un análisis inicial en cada iteración (sprint).

Podría deducirse que SCRUM puede resultar más lenta que XP al poder provocar una revisión en cada nueva iteración, pero hay que tener en cuenta que una vez se ha aprobado un cambio ya no se vuelve a trabajar con él. Esto no ocurre en XP, ya que las tareas que se van completando en anteriores entregas pueden modificarse con posteridad aunque funcionen correctamente.

Tras este análisis y teniendo en cuenta nuestra forma de trabajo y el proyecto a abordar, decidimos aplicar la técnica de XP. Pese a ser una metodología válida, SCRUM nos pareció más útil para equipos más amplios, facilitando la administración del proyecto. Al trabajar en equipo XP resultó más acorde, ya que fue concebido para trabajar en parejas, mientras que SCRUM resulta más individualista. El factor determinante fue sin duda la flexibilidad en las modificaciones. Al trabajar con una plataforma tan poco explorada, no descartamos la posibilidad de que tal vez tuviéramos que dar marcha atrás en algún momento o cambiar algún elemento ya creado para acomodar otro.

Características de XP

Los cuatro valores fundamentales de toda metodología ágil son: comunicación, simplicidad, coraje y retroalimentación. Además de estos valores, que podrían considerarse los 4 pilares de XP, esta metodología también nos introduce 12 valores añadidos en un manifiesto propio, que describimos a continuación (Newkirk, J. y Martin, R. C., 2001):

- **Planificación:** El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.
- **Entregas pequeñas:** La idea es producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad pretendida para el sistema. Una entrega no debería tardar más de 3 meses.
- **Metáfora:** En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente se solventan con una metáfora. Entendemos por metáfora una historia compartida que describe cómo debería funcionar el sistema.
- **Diseño simple:** Se debe diseñar la solución lo más simple posible de implementar en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente.
- **Pruebas:** La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema.
- **Refactorización (*Refactoring*):** La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios.
- **Programación en parejas:** Toda la producción de código debe realizarse con trabajo en parejas de programadores, se ha demostrado que el tiempo de desarrollo se ve reducido y la calidad del código resultante aumentada (4:49-50).
- **Propiedad colectiva del código:** Cualquier programador puede cambiar cualquier parte del código en cualquier momento.
- **Integración continua:** Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- **40 horas por semana:** Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.
- **Cliente in-situ:** El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada.

- **Estándares de programación:** XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados).

Se consigue un mayor beneficio al aplicar las anteriores prácticas de forma conjunta y equilibrada puesto que se apoyan unas en otras. Esto se ilustra en la siguiente ilustración (obtenida de "*Extreme Programming Explained. Embrace Change*" (Wake, W. C., 2002) junto con los valores expuestos), donde una línea entre dos prácticas significa que las dos prácticas se refuerzan entre sí.

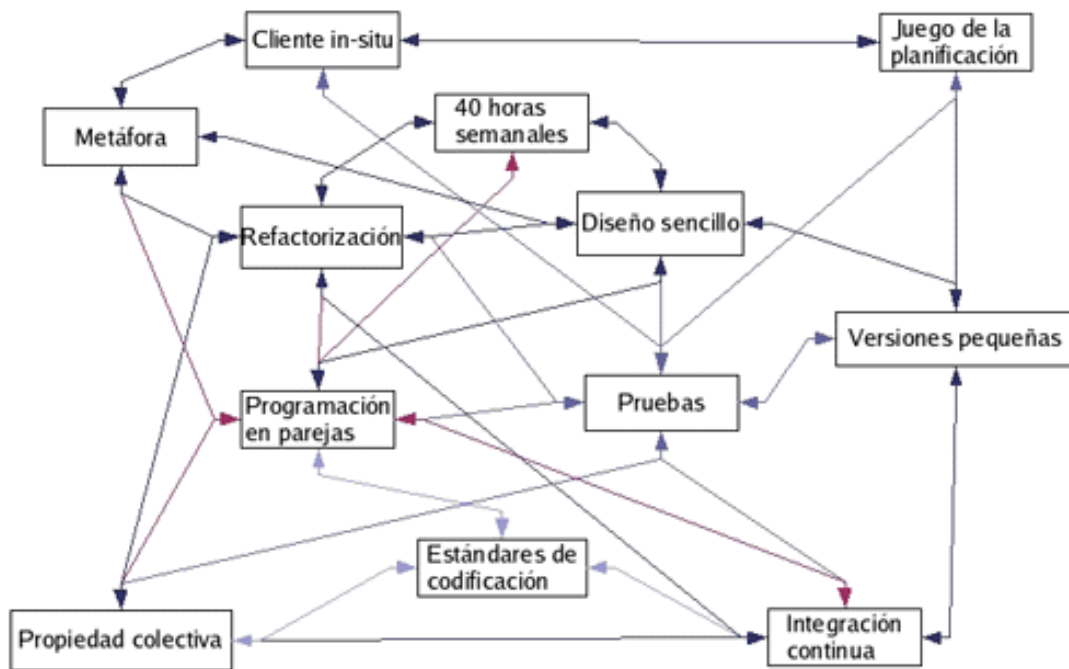


Ilustración 2 - Prácticas XP

Debido a la naturaleza del proyecto a abordar, la documentación y el control de versiones va a ser un factor muy importante. No solo por las continuas pruebas a las que vamos a someter al mismo, sino también por el entorno que vamos a vernos forzados a utilizar. A estas alturas, ya hemos repetido varias veces lo novedoso de la plataforma. Esto tiene sus ventajas e inconvenientes, entre ellos el que tal vez nos tengamos que ver forzados a deshacer cambios o modificar características a lo que la plataforma nos permite contra lo que teníamos pensado. Un control férreo de los cambios en el código, tanto por evolución como por necesidad, va a ser necesario de cara al éxito del proyecto.

La retroalimentación constante, tanto con los usuarios como con interesados en *analizo.info*, es otro valor añadido a este proyecto ya que sus impresiones, opiniones y conclusiones en las pruebas impulsarán el desarrollo, además de apoyar las políticas de simplicidad y comunicación que aspiramos a utilizar.

Como es lógico en todo desarrollo, nos dedicaremos a hacer un análisis previo antes de entrar de pleno en el proyecto. Para ello intentaremos hacer una breve descripción de a qué nos enfrentamos, intentando obtener una perspectiva inicial lo más completa posible.

Es a partir de este momento cuando vamos a tener que empezar a trabajar con el ciclo iterativo de XP. Con cada nueva *release* o cambio implementado, se realizarán pruebas de usuario, que a su vez realimentarán la próxima versión hasta así obtener un resultado final. En la siguiente ilustración mostramos la teoría, aplicada a nuestro proyecto, detrás de las iteraciones de XP.

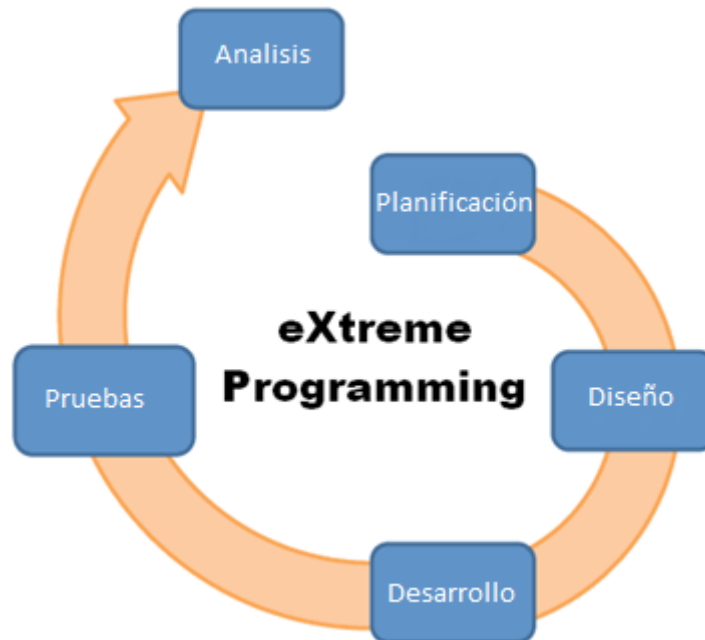


Ilustración 3 - Estructura eXtreme Programming

A partir de un resultado inicial se realiza una planificación de cambios o correcciones a implementar. En este documento nos centraremos más en el núcleo y codificación de la aplicación dado que el diseño se va a abordar en un proyecto paralelo. Es por ello que conjuntamente y en pareja, se realizará un diseño preliminar de los cambios a implementar, tanto en el aspecto visual como en el práctico. Estos cambios serán posteriormente desarrollados por ambas partes. Tras lo cual, los cambios serán sometidos a pruebas, tanto por parte de *analizo.info* como de público independiente. Estas pruebas pasarán a ser analizadas y estudiadas para así poder iniciar la próxima iteración.

Organización del equipo del proyecto

Como ya se ha comentado con anterioridad en este documento, el equipo está formado por dos desarrolladores. En este proyecto se llevará un control de la funcionalidad y del núcleo de código, mientras que en el proyecto paralelo “*DESARROLLO DE APLICACIÓN INTERACTIVA PARA TABLET Y SMARTTV EN UN ENTORNO "SECOND SCREEN - (DCADHA-14)*”, se hará uso de un enfoque más inclinado hacia la experiencia de usuario.

La retroalimentación tiene un fuerte impacto en ambas partes, pero especialmente en éste segundo proyecto. La primera cualidad que alguien percibe de una aplicación es su diseño y aspecto visual, es el factor más susceptible a cambios. No

necesariamente porque los necesite pero porque es un aspecto obvio en el que resulta sencillo detectar mejoras o flaquezas, justamente porque es lo primero que ve el usuario. Es por ello por lo que se hará uso de técnicas de desarrollo y diseño como es *User Centered Design* (UCD).

Según wikipedia, “*El Diseño Centrado en el Usuario es una filosofía de diseño que tiene por objetivo la creación de productos que resuelvan necesidades concretas de sus usuarios finales, consiguiendo la mayor satisfacción y mejor experiencia de uso posible con el mínimo esfuerzo de su parte.*”

La mayoría de proyectos que aplican UCD cumplen las siguientes características:

- Conocer a fondo a los usuarios finales
- Diseñar un producto que resuelva sus necesidades y se ajuste a sus capacidades, expectativas y motivaciones
- Poner a prueba lo diseñado, normalmente usando test de usuarios

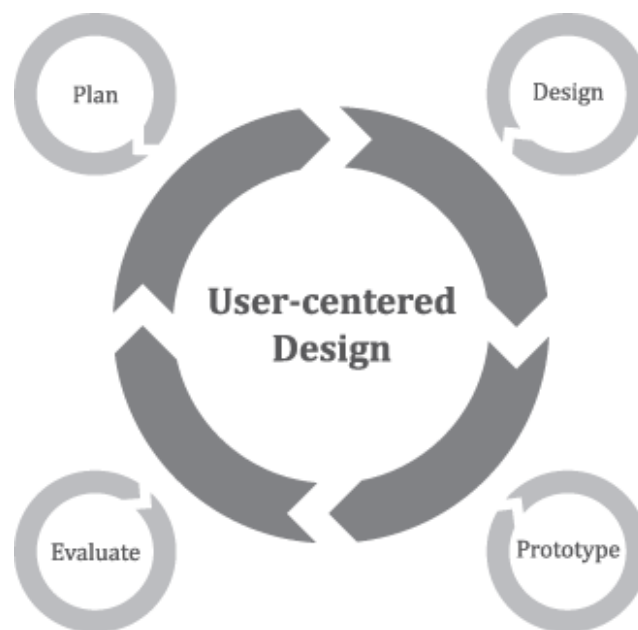


Ilustración 4 - Estructura UCD

Como podemos comprobar en la anterior ilustración, la metodología de trabajo no difiere mucho de XP, cualidad que nos facilita muchísimo el realizar un desarrollo en paralelo de ambas partes. A grandes rasgos ambas partes del proyecto pasan por las mismas fases: Planificación, diseño, implementación, testeo y análisis de resultados. De modo que mientras ambos trabajaremos aspectos diferentes de la misma aplicación, lo haremos en todo momento de forma paralela con el fin de crear versiones lo más completas posibles con cada iteración, mejorando de forma conjunta distintos aspectos de la misma. Con esta metodología, también pretendemos reducir lo máximo posible el número de iteraciones, abarcando lo máximo posible dentro de cada una.

Herramientas de desarrollo y colaboración utilizadas

Teniendo en cuenta la plataforma para la cual vamos a desarrollar, se va a trabajar con el entorno de programación *Eclipse*, incluido en el SDK de Samsung y configurado para trabajar con él.

Como se ha indicado con anterioridad, haremos uso de lenguajes ampliamente conocidos como son *HTML5* y *Javascript/jQuery*.

Samsung utiliza una serie de funciones escritas en *Javascript* que controlan los eventos que se producen en el sistema. Así podemos diferenciar tres capas:

- **Navegador:** Dónde se visualiza el contenido y se ejecutan las funciones en Javascript. Este hace llamadas a la API.
- **Application Programming Interface (API):** Que hace de intermediario entre el navegador y los eventos del sistema. Es invocada por el navegador.
- **Sistema Operativo o gestor de eventos:** Que es el que realmente permite que se lleven a cabo operaciones importantes como ver la televisión o cambiar de canal.

Además dentro del SDK de Samsung, como se puede ver en SDK download, Guide & Forum (2013), existen tres tipos de proyectos que pueden implementarse, pasamos a resumirlos a continuación:

- **Básicos:** En *JavaScript*, con objetos propios y librerías propias del SDK. Además proporciona a los desarrolladores la oportunidad de crear su aplicación utilizando el Editor de Visual. Funciona con el concepto de que una aplicación se divide en escenarios o vistas.
- **JavaScript:** En *JavaScript*, libertad de utilizar sus librerías o propias del usuario. Es más similar a un desarrollo web clásico, ofrece la posibilidad de trabajar a bajo nivel, dando más libertad al desarrollador pero añadiendo a la vez más complejidad a la hora de programar.
- **Flash:** En *ActionScript 2.0*. *Samsung SmartTV* es compatible con el uso de funciones Flash, permite el uso de las funciones estándar de Flash así como la implementación de estos componentes de acuerdo con los modelos previstos.

Nosotros nos hemos decantado por un proyecto básico porque nos permite usar las librerías nativas y controlar la señal de televisión entre otras cosas. Para aquellos que no están familiarizados con el concepto de escenarios, es un principio sencillo de entender. Un escenario es un área rectangular en la que se coloca el contenido gráfico y la aplicación actúa a forma de marco. De este modo, podemos tener varios escenarios dentro de un mismo marco, ocultando o superponiendo los que nos interesen en cada momento. Nuestra intención no era la de realizar una aplicación en flash por lo que esta opción quedo descartada, y pese a la flexibilidad que ofrecía un proyecto *JavaScript*, debido a las funciones que pretendíamos usar

un proyecto básico se ajustaba mucho más a nuestras necesidad facilitándonos la implementación y uso de las mimas.

Persistencia de la información

La iniciativa *analizo.info* utiliza un sistema clásico de base de datos (BBDD) *MySQL* pero *Javascript* no es compatible de manera nativa con esta tecnología. Es por eso que nos hemos decantado por hacer uso de las cada vez más populares *Application Programming Interfaces* (API), que nos permitirán almacenar la información de forma remota sin tener que ejecutarse en el mismo servidor. De esta forma, en un futuro, los datos serán accesibles desde cualquier plataforma y por los usuarios que queramos. Esta solución no solo nos permite crear una aplicación multiplataforma, de cara al servidor que la aloja, sino también ofrece la posibilidad de un futuro ampliar su funcionalidad y características de forma modular sin tener que actuar necesariamente sobre el propio código o la BBDD.

Para llevarlo a cabo nos hemos decantado por <https://apigee.com>. *Apigee* es un servicio que nos permite tener nuestra API accesible en la nube de forma gratuita y crear nuestros propios modelos de datos. Además cuenta con librerías *JavaScript* por lo que queda integrado con nuestro código. Al usar una API ajena esta ofrece ciertas limitaciones, como por ejemplo el control de correo. Sin embargo, en nuestro caso y debido a las aspiraciones de este proyecto esta solución resultó más que suficiente.

En un futuro lo ideal sería contar con una API propia. Existen opciones como *django* en *Python*, pero nosotros nos decantamos por *Node.js* que nos parece una herramienta mucho más modular y potente. El hecho de construir una API propia podría ser motivo de un nuevo proyecto debido al coste de recursos y tiempo necesarios para llevarlo a cabo. Debido a las naturales limitaciones de tiempo, nos vemos obligados a hacer uso del servicio *apigee* que nos proporciona todo lo que necesitamos.

No hemos querido ahondar en este punto a fin de no eclipsar le resto del contenido, sin embargo consideramos que es un factor importante de ser estudiado en futuros proyectos o mejoras del presente proyecto. Por ello, quien así lo desee, puede revisar la información relativa a *Apigee* y *Node.js* que manejamos durante el transcurso de este proyecto en el Anexo III.

Control de los datos

En una aplicación de estas características, que hace uso tecnología web y se ejecuta en un entorno poco convencional, es necesario construir una estructura de datos que no dependa de la misma aplicación. En este caso nos decidimos por una tecnología de API REST que pueda ser útil en futuros proyectos ejecutados en terminales móviles.

A continuación pasamos a resumir gráficamente la estructura de la API:

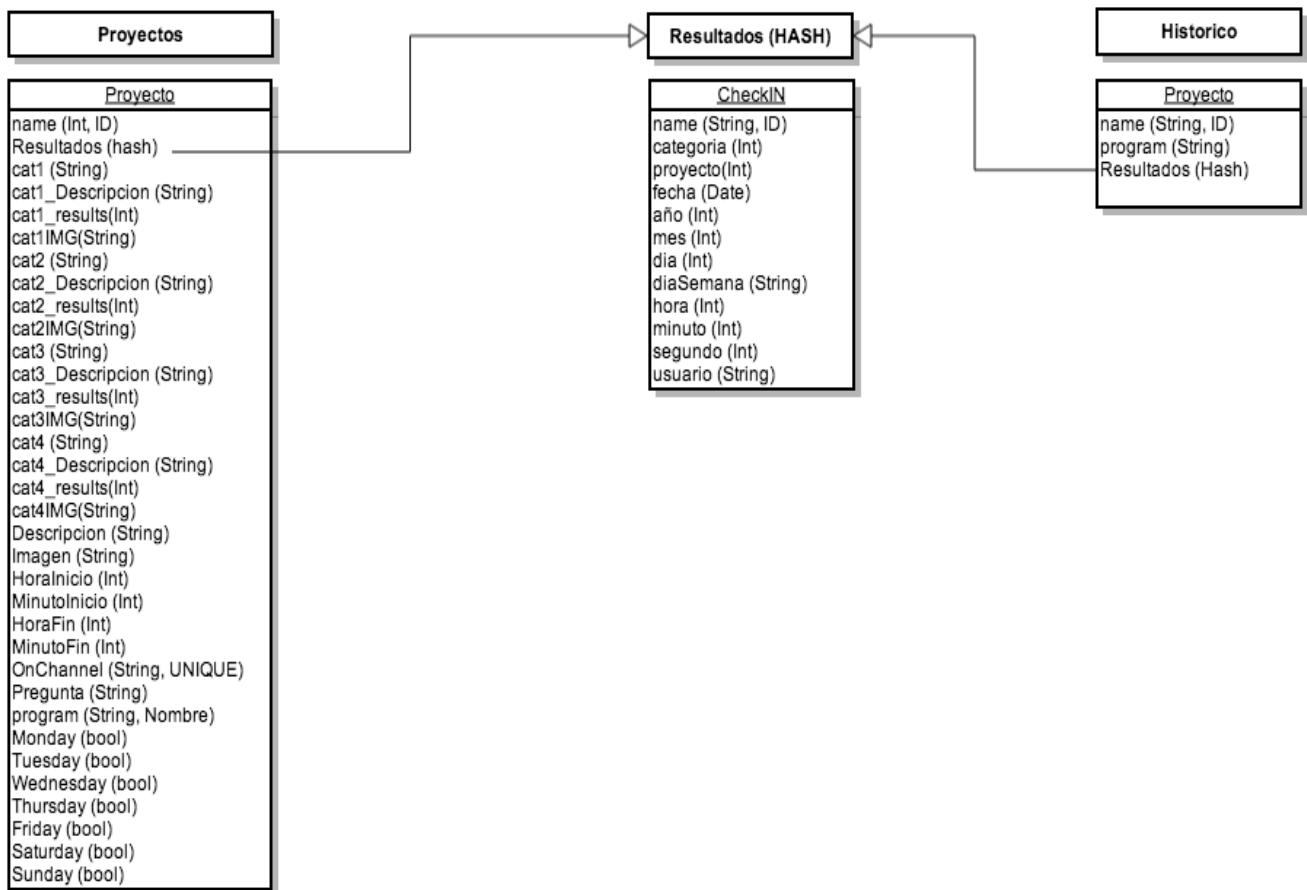


Ilustración 5 - Estructura de la API

Como hemos dicho anteriormente, la API que utilizamos está alojada en el servicio *apigee.com* y se accede a través de peticiones GET/POST devolviendo un objeto JSON. Como podemos ver en la ilustración anterior, contamos con tres “estructuras” involucradas: la referente a los proyectos (con sus elementos y variables), una estructura que contiene un histórico de los resultados ya registrados y la estructura encargada de registrar nuevos resultados (opciones enviadas).

No hace falta especificar que entre todas hay una interrelación, cada una de ellas comparte datos con las demás, la nueva votación hará uso de los datos proporcionados por el proyecto, así como la nueva votación será almacenada en el histórico con datos adicionales. Cabe destacar que en caso de borrado de un proyecto específico, los datos de análisis no son borrados. Por el contrario, añadimos un nuevo objeto al historial de forma que estos puedan ser consultados y no se pierdan los anteriores.

Control de cambios

Para poder llevar el control de cambios del proyecto se ha optado por hacer uso de la herramienta *GitHub*, que utiliza el sistema de control de versiones *Git*.

Este sistema nos ofrece llevar la posibilidad de llevar un control online, además de ofrecernos una aplicación nativa y multiplataforma para sincronizar nuestro contenido y llevar un control de los cambios en cualquier momento.

El sistema tiene ciertas limitaciones, como por ejemplo el tamaño máximo de repositorio de 1Gb y la libre disponibilidad del código subido (versión gratuita). Debido a la naturaleza de nuestro proyecto y la filosofía de *analizo.info*, ninguno de estos factores supone un inconveniente, al ser nuestro proyecto de tamaño reducido y además estar a favor del código abierto.

El uso de diferentes cuentas y perfiles que nos proporciona *GitHub* además permite mejorar nuestra comunicación y testing, facilitando a cualquier persona relacionada con el proyecto formular preguntas, documentar bugs o sugerir mejoras a través de la propia plataforma.

Parte del código desarrollado y revisiones específicas serán analizadas con detalle en posteriores etapas de este documento. No obstante, todos los cambios por los que ha pasado el proyecto, así como el código fuente del mismo se pueden consultar en:

<https://github.com/analizo-develop/PFC-AnalizoTV>

The screenshot displays the GitHub interface for the repository 'analizo-develop / PFC-AnalizoTV'. At the top, there's the GitHub logo, a search bar, and navigation links for 'Explore', 'Features', 'Enterprise', and 'Blog'. There are 'Sign up' and 'Sign in' buttons. Below the repository name, it shows 'PUBLIC' and 'analizo-develop / PFC-AnalizoTV' with '2' stars and '0' forks. The main content area shows a progress bar for 'JavaScript 80.1%' and 'CSS 19.9%'. Below that, there's a table of files and folders with their commit history. The right sidebar contains links for 'Code', 'Issues', 'Pull Requests', 'Wiki', 'Pulse', 'Graphs', and 'Network'. At the bottom right, there are buttons for 'Clone in Desktop' and 'Download ZIP'.

File/Folder	Commit Message	Time Ago
minor bugfixes		
analizo-develop	authored a month ago	latest commit b224129478
app	minor bugfixes	a month ago
icon	UI Improvements, errors managed, changed logo	2 months ago
images	minor bugfixes	a month ago
layout	minor bugfixes	a month ago
.gitattributes	First Commit	3 months ago
.gitignore	First Commit	3 months ago
app.json	Added new Info Screen (Left in Login)	a month ago
config.xml	First Commit	3 months ago
index.html	UI Improvements, errors managed, changed logo	2 months ago
widget.info	First Commit	3 months ago

Ilustración 6 - Panel de control Github

Planificación

A fin de asegurar un desarrollo fluido hemos procedido a hacer un análisis preliminar de la aplicación. Como ya hemos explicado, XP es partidario de una planificación más bien laxa, sin entrar en detalles. A partir de un concepto inicial y a través de las iteraciones necesarias, se consigue que el concepto vaya tomando forma hasta obtener una especificación final.

Ahora bien, todas las cualidades que hacen de la XP una metodología eficaz de programación, se vendrían abajo si además no contásemos con unos mínimos estándares de codificación. Es necesario para el buen término del proyecto establecer unos estándares de forma que los desarrolladores sigan los mismos criterios para desarrollar código. Esto no solo dota de eficacia al tomar decisiones para satisfacer los requerimientos propuestos por el cliente, sino también ayuda y facilita la comunicación entre los miembros del equipo (Letelier, P. y Sánchez, E., 2003:25).

Podemos afirmar pues, que pese a que la parte de análisis es mucho más ligera en XP, la parte de diseño sigue siendo parecida a las labores de diseño más o menos habituales en otras metodologías. Una vez todos los miembros del equipo han comprendido lo que se va a realizar, la primera parte del análisis está completa. En posteriores etapas se realizará un análisis más específico y detallado dependiendo de la iteración.

La metodología XP en ningún momento especifica que esta etapa de análisis deba ser documentada o modelada, toma una posición mucho más abierta, dejando libertad a los desarrolladores para usar o no herramientas de modelado, siempre y cuando estas vayan a servir de ayuda para la comunicación entre el equipo de desarrollo.

En nuestro caso no entraremos en excesivo detalle con diagramas como el de secuencia, colaboración o clases. No obstante, a fin de facilitar la lectura y la comprensión global al equipo de desarrollo, hemos hecho uso de un diagrama con *Business Process Management Notation* (BPMN).

BPMN no es una herramienta comúnmente asociada a XP, sin embargo hemos considerado oportuno su inclusión ya que refleja los procesos envueltos de forma clara, además de ser una notación comúnmente utilizada para la comunicación entre desarrolladores y clientes, justamente por su fácil comprensión.

A fin de facilitar el control de cambios, haremos uso de casos de uso frente a los comúnmente conocidas historias de usuarios de XP. Esta decisión puede que hubiera violado la política ágil de XP en un proyecto de mayor escala, pero dado la naturaleza de nuestro proyecto no nos suponía un sobreesfuerzo adaptar las historias de usuarios a casos de uso.

Una historia de usuario describe funcionalidad que será útil para el usuario de un sistema software. A grandes rasgos contiene la información extraída de las conversaciones con el cliente. Una historia de usuario debería ser pequeña, memorizable, y que pudiera ser desarrollada por un par de programadores en una semana. Pero al tener como mucho un par de frases, es imposible que una historia

de usuario contenga toda la información necesaria para desarrollar. En tan reducido espacio no se puede reflejar el diseño, las pruebas, normativa,...etc.

Este es uno de los principales problemas que suelen encontrar los equipos de desarrollo a la hora de trabajar con metodologías ágiles. En la bibliografía consultada, podemos encontrar diversas opiniones con respecto a utilizar casos de uso frente a historias de usuario (Cohn, M., 2004; Wake, W. C, 2002). Tras un estudio de las diferentes posiciones, casi todos coinciden que un caso de uso puede almacenar información acerca de un cliente que una historia de usuario no, así como que una historia de usuario fuerza una conversación con el cliente, mientras que el caso de uso es un análisis posterior a dicha conversación.

En ningún momento ninguna de la documentación utilizada toma una posición frente a la otra, todo lo contrario ya que es una conducta habitual. No obstante hay que tener en cuenta que son herramientas distintas y útiles en función de su uso. Un caso de uso solo será útil a un proyecto XP, únicamente si es actualizado hasta su finalización, y no cuando se usan al estilo tradicional para definir lo que hay que construir sin posibilidad de cambios posteriores (Fowler, M., Foemmel M., 2001).

Hay que entender que las historias de usuario son lo que son porque su objetivo es, entre otros, lograr la interacción con el equipo y con el usuario. Aunque la práctica haga que en ocasiones la teoría se deba ajustar a la práctica, para lo cual hay quien necesariamente relaja la agilidad para aplicar técnicas tradicionales en ciclos de vida ágil. Hay quien usa casos de uso en vez de la historia de usuario o hay quien utiliza técnicas de trazabilidad para relacionar la historia de usuario con otros documentos. En conclusión, la teoría ágil se debe adaptar a cada caso en particular, muchas veces relajando la agilidad, a fin de obtener la verdadera riqueza de las múltiples soluciones que ofrece la ingeniería del software (Cohn, M., 2004).

A continuación procederemos a hacer un análisis superficial a través de un diagrama de procesos BPMN. Posteriormente desglosaremos dichos procesos en casos de uso y finalmente realizaremos una planificación temporal estimativa.

Diagrama de procesos

A continuación se muestra el *Business process diagram* (BPD). Esta notación se caracteriza por ser más abstracta que la mayoría de herramientas de modelado. Sin embargo, es una notación muy sencilla de entender, tanto para personal técnico como ajeno a la industria. Se utiliza para modelar gráficamente las operaciones de los procesos, de forma que los usuarios con reducido conocimiento técnico puedan leer y comprender los procesos sin ningún problema.

En el siguiente gráfico se puede apreciar el funcionamiento general de la aplicación, se contemplan todas las operaciones y posibilidades de una forma generalizada. El usuario empieza iniciando sesión si ya tiene datos de registro, en caso contrario deberá pasar por un proceso de registro. Tras estos los datos son cotejados y almacenado en *Apigee*, arrojando los errores oportunos en caso de que estos no sean correctos.

Una vez se ha iniciado sesión tendríamos una vista general con los proyectos disponibles donde seleccionaríamos un proyecto específico. Esto solo podrá realizarse para aquellos proyectos que se encuentren dentro de su horario predefinido, en caso contrario se deberá seleccionar otro proyecto. Una vez dentro de un proyecto se podrán enviar cuantas votaciones el usuario considere oportunas. El usuario podrá salir en cualquier etapa del proceso y cómo podemos comprobar, todo lo que suponga intercambio de información se notifica y registra en *Apigee*.

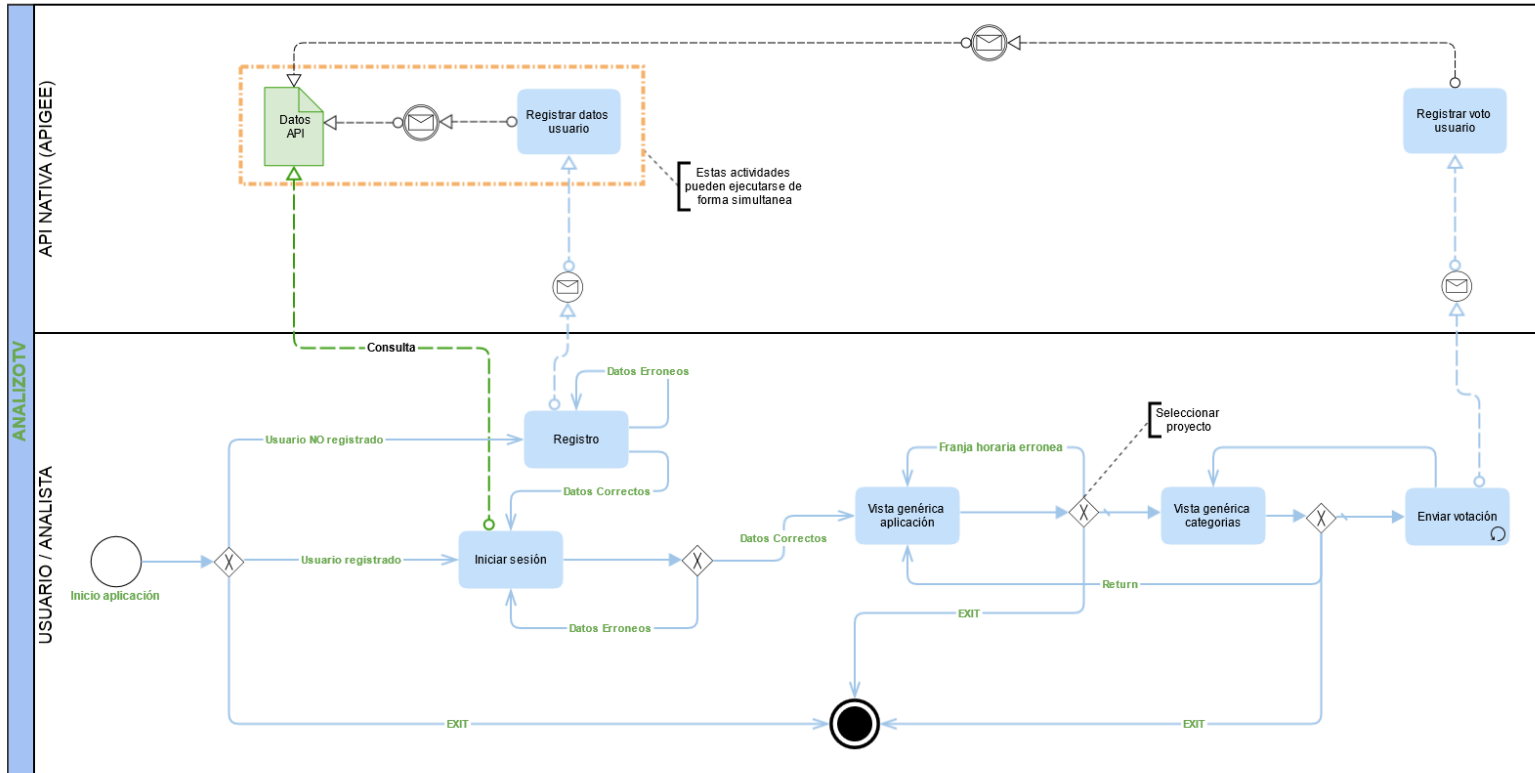


Ilustración 7 - Diagrama BPMN

Hay que tener en cuenta que este diagrama es una versión inicial del proyecto, ideada en su concepción para obtener una idea global y de uso interno. Es por este motivo, por lo que a lo largo del proyecto puede haber cambios respecto a esta concepción inicial.

Tras el análisis de este diagrama, así como los casos de uso expuestos a continuación, se tendrá una imagen completa y certera de que es lo que esperamos obtener del desarrollo de este proyecto.

Casos de uso

A continuación pasaremos a detallar las interacciones que podremos encontrar en nuestro sistema mediante casos de uso UML. En ellos podremos ver los actores involucrados, así como las posibles acciones y reacciones.

El diagrama de procesos, anteriormente expuesto, nos otorga una visión global del problema. Mediante la especificación de casos de uso lo que pretendemos es focalizar más la atención en pequeños detalles e interacciones que no siempre podemos captar en un diagrama. Como ya hemos comentado esto supone una variación a las ya conocidas historias de usuario, pero creemos que esta solución supone una buena alternativa y, al menos en este particular, nos resultará útil a lo largo de todo el proyecto.

Los casos de uso expuestos a continuación no fueron los concebidos inicialmente, muchos, por no decir todos, han sido ampliados y modificados. Algunos de ellos se añadieron con posteridad y todos han sido adaptados a las especificaciones de *analizo.info* en un momento u otro.

El uso de UML como lenguaje de especificación es complicado y en ocasiones se hace difícil de leer hasta para los propios desarrolladores, por lo que procederemos a analizar los casos de uso de una forma más teórica para posteriormente representar las interacciones expuestas de forma más gráfica.

El siguiente caso de uso pasa a describir la primera pantalla que ve el usuario al iniciar la aplicación. Este caso en particular fue añadido en etapas posteriores del desarrollo, su finalidad es la de permitir la correcta carga de los elementos *JavaScript* en un segundo plano, mientras esto ocurre, al usuario se le muestra una común pantalla de splash.

CU-01: Pantalla de Inicio

Resumen	<i>Primera pantalla visible al usuario nada más arrancar la aplicación. Es una pantalla de presentación con el nombre de la aplicación, visible al usuario por apenas unos 5 segundos.</i>
Importancia	MEDIA – No solo permite al usuario ambientarse en la aplicación, sino que sus 5 segundos de duración asegura que las librerías Javascript/jQuery se carguen de forma adecuada.
Frecuencia de uso	BAJA – El usuario solo verá esta pantalla una vez cada vez que ejecute la aplicación.
Actores principales	USUARIO
Acciones	<ol style="list-style-type: none"> 1. El usuario arranca la aplicación en la SmartTV 2. La aplicación muestra la pantalla de Inicio al arrancar 3. Durante la visualización de la pantalla de inicio el resto de la aplicación se carga en un segundo plano 4. La pantalla de inicio desaparece y da paso a la pantalla de login si el usuario no ha iniciado sesión o esta a caducado.
Error	Cualquier error en este elemento será debido a una excepción HW/SW. No hay ningún posible error controlado.
Excepciones	La aplicación almacenará una cache de datos, por lo que si los datos de registro de un usuario están disponibles, la pantalla de inicio dará paso directamente a la aplicación

Tabla 2 - Caso de Uso 1

La siguiente pantalla es accesible desde la pantalla de login de la aplicación. Su función es la de familiarizar al usuario con la política y filosofía de *analizo.info*. Como en el caso anterior, esta pantalla fue un requerimiento de la plantilla de *analizo.info* y por lo tanto, añadido con posteridad.

CU-02: Pantalla de información

Resumen	<i>Esta pantalla sirve para familiarizar al usuario con la política y filosofía de analizo.info. Su finalidad es la de hacer una breve introducción de la naturaleza de la aplicación, por quien está desarrollada y para sirve.</i>
Importancia	BAJA - Esta pantalla no tiene una funcionalidad real. El usuario no pierde ningún dato relevante por no visitarla, sin embargo ofrece información adicional a aquellos usuarios interesados
Frecuencia de uso	BAJA – Esta pantalla se visitará entre ninguna y 1 vez por cada usuario. Es posible que se utilice como consulta en algún otro momento, pero la relación de visitas con respecto al resto de la aplicación será mínima.
Actores principales	USUARIO
Acciones	<ol style="list-style-type: none"> 1. Desde la pantalla de login, el usuario accede a la opción “Acerca de..” 2. Una vez dentro de la pantalla, se explicará en texto plano la filosofía de <i>analizo.info</i>, pero ante todo la naturaleza y funcionalidad de la aplicación 3. Una vez el usuario está satisfecho con la lectura, se volverá a la pantalla de inicio de sesión
Error	Cualquier error en este elemento será debido a una excepción HW/SW. No hay ningún posible error controlado.
Excepciones	Esta pantalla solo será accesible desde la pantalla de inicio de sesión, por lo que si un usuario quiere consultar una vez dentro de la aplicación, este deberá cerrar sesión.

Tabla 3 - Caso de Uso 2

Esta pantalla tiene como función la identificación del usuario para entrar a los servicios de la aplicación. De todos los casos contemplados, es el caso que menos alteraciones ha sufrido y que estuvo presente desde el inicio del desarrollo.

CU-03.1: Pantalla de Login

Resumen	<i>Para poder acceder a la aplicación todo usuario deberá identificarse. Para ello deberá haber sido dado de alta en la API implementada a fin de administrar los proyectos. En esta todo usuario deberá facilitar su nombre y contraseña de acceso.</i>
Importancia	ALTA – Sin introducir los datos correctos en esta pantalla ningún usuario quedará registrado en el sistema. El acceso a la aplicación NO es posible sin superar esta pantalla.
Frecuencia de uso	BAJA – El usuario solo verá esta pantalla la primera vez que inicie sesión.
Actores principales	USUARIO y API
Acciones	<ol style="list-style-type: none"> 1. Al usuario se le muestra pantalla de acceso con campos editables para introducir usuario y contraseña 2. El usuario debe introducir datos de acceso 3. Cuando el usuario envía los datos la aplicación los coteja contra la API 4. Si el resultado es satisfactorio el usuario accede a la pantalla principal de la aplicación 5. Si el resultado es erróneo se le mostrará mensaje de error
Error	El usuario puede introducir mal los datos (Usuario inexistente, usuario erróneo y/o contraseña errónea). Cuando esto ocurra, la aplicación mostrará un dialogo emergente informando al usuario de los datos erróneos. Esta ventana desaparecerá por si sola a los 2 segundos.
Excepciones	Si el usuario ha iniciado sesión con anterioridad y no ha cerrado sesión, en el próximo acceso no se le solicitarán los datos, la aplicación guarda el token de acceso (cookies).

Tabla 4 - Caso de Uso 3.1

Al igual que la pantalla anterior, este caso fue definido en la concepción de la aplicación. Sus alteraciones han sido mínimas y tiene como finalidad la de registrar un nuevo usuario en *Apigge*, para que este a su vez pueda iniciar sesión en la aplicación si así lo desea.

CU-03.2: Pantalla de registro

Resumen	<i>En caso de que un usuario no esté registrado, este no podrá iniciar sesión y por consiguiente, no podrá acceder a la aplicación. Se le da la posibilidad de registrarse a cualquier usuario desde la propia aplicación.</i>
Importancia	ALTA – Todos los usuarios deberán pasar al menos una vez por esta pantalla para poder obtener su cuenta de analista y poder utilizar la aplicación.
Frecuencia de uso	BAJA – Esta pantalla solo será utilizada por aquellos usuarios sin datos de acceso.
Actores principales	USUARIO y API
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a esta pantalla desde la pantalla de inicio de sesión común. 2. El usuario debe introducir datos de acceso, como son: nombre de usuario, contraseña y dirección de correo 3. Cuando el usuario envía los datos la aplicación los coteja contra la API 4. Si el resultado es satisfactorio al usuario se le muestra mensaje emergente notificándole del correcto registro 5. Si el resultado es erróneo se le mostrará mensaje de error 6. Una vez el usuario esté registrado se le devuelve a la pantalla de inicio de sesión para acceder a la aplicación
Error	El usuario puede introducir mal los datos (Usuario ya existente, dirección de correo ya registrada y/o contraseña mal introducida). Cuando esto ocurra, la aplicación mostrará un dialogo emergente informando al usuario de los datos erróneos. Esta ventana desaparecerá por si sola a los 2 segundos.
Excepciones	Esta pantalla solo será accesible desde la pantalla de inicio de sesión, por lo que si un usuario quiere consultar una vez dentro de la aplicación, este deberá cerrar sesión.

Tabla 5 - Caso de Uso 3.2

Al igual que las pantallas anteriores, este caso está presente desde el arranque de nuestro proyecto. Pero a diferencia de los anteriores casos, esta pantalla ha sufrido cambios drásticos a través de las iteraciones. Su finalidad es la mostrar todos los proyectos disponibles, para que de este modo, el usuario elija el que más le interese. Su distribución, diseño y características han cambiado desde su concepción.

CU-04.1: Pantalla de aplicación

Resumen	<i>Una vez el usuario acceda a la aplicación, esta será la primera pantalla que verá. Se trata del núcleo de la aplicación, ya que en ella se pueden consultar todos los proyectos disponibles, así como su disponibilidad horaria y su temática. Desde esta pantalla se puede acceder a los proyectos que cumplan la restricción horaria. Aquellos que no se encuentren dentro de su horario se mostrarán en la misma pantalla por separado, pero sin posibilidad de seleccionarlos.</i>
Importancia	ALTA – Toda la funcionalidad de la aplicación se contempla en esta pantalla. Muestra al usuario los proyectos a su disposición, además muestra claramente y por separado los proyectos fuera de la franja horaria efectiva.
Frecuencia de uso	ALTA - Como ya hemos dicho se trata del núcleo de la aplicación, el usuario pasará por esta pantalla para moverse entre proyectos. Una vez elegido un proyecto no es necesario volver a ella pero si lo es si queremos cambiar de proyecto.
Actores principales	USUARIO y API
Acciones	<ol style="list-style-type: none"> 1. Una vez el usuario ha sido identificado acceder a la pantalla de aplicación 2. El usuario deberá seleccionar un proyecto de entre las aplicaciones disponibles 3. Las aplicaciones disponibles están alojadas en la API, su disponibilidad esta cotejada según franja horaria. Si un usuario está interesado en un proyecto que todavía no es disponible, puede salir de la aplicación y volver más tarde. 4. Una vez un usuario selecciona un proyecto, será llevado a la pantalla específica de dicho proyecto activando el cambio de canal.
Error	Cualquier error en este elemento será debido a una excepción HW/SW. No hay ningún posible error controlado.
Excepciones	Ninguna contemplada

Tabla 6 - Caso de Uso 4.1

La finalidad de esta pantalla es la mostrar los distintos votos disponibles dentro de un proyecto. Una vez se accede a un proyecto, esta pantalla será la encargada de enviar los votos para su registro en *Apigee*. Pese a ser una pantalla definida en las etapas iniciales, los cambios que ha sufrido, han sido sobre todo estéticos.

CU-04.2: Pantalla de proyecto

Resumen	<i>Esta pantalla en realidad se trata de una prolongación de CU-04.1. Esta se genera dinámicamente. Cuando un usuario selecciona un proyecto tiene a su disposición 4 posibles votos que enviar. Son votos preestablecidos, a fin de actuar como estadísticas de eventos específicos en <i>analizo.info</i>. Cuando un usuario considera que se da un evento, bastará con que seleccione la opción para dicho voto y este se registrará en la API.</i>
Importancia	ALTA – Todas las votaciones deberán pasar antes por esta pantalla, una vez un usuario ha accedido a esta pantalla nos aseguramos que se ha controlado su identidad, la franja horaria, así como el canal. Una vez enviamos un voto, este ha sido restringido a su especificación y puede ser enviado para su registro.
Frecuencia de uso	ALTA – Desde esta pantalla el usuario puede enviar varios votos, tanto como le sea posible la duración del programa. Sin duda esta será la pantalla, sino más utilizada, si la que más tiempo permanecerá en pantalla.
Actores principales	USUARIO y API
Acciones	<ol style="list-style-type: none"> 1. Habiendo seleccionado un proyecto, al usuario se le muestra la pantalla de proyecto con los 4 votos disponibles 2. Una vez el usuario detecta algún evento que cumple los criterios de uno o varios votos, este procederá a pulsar la tecla correspondiente al voto 3. La selección se enviará a la API, donde quedará registrada 4. Durante el proceso de envío se le mostrara una ventana emergente al usuario, donde se le indicará que su voto está siendo registrado. La ventana desaparecerá una vez el voto haya sido enviado con éxito
Error	Cualquier error en este elemento será debido a una excepción HW/SW. No hay ningún posible error controlado.
Excepciones	Si el usuario se encuentra dentro de un proyecto y se dispone a envía un voto una vez finalizada la franja horaria de un proyecto, la aplicación le mostrará una ventana emergente informándole de la imposibilidad de registrar el voto. El aviso desaparecerá por si solo a los 2 segundos.

Tabla 7 - Caso de Uso 4.2

La siguiente pantalla es accesible antes de entrar a la pantalla del proyecto. Su función es la de servir al usuario de ayuda a la hora de utilizar la aplicación. Se hace un breve resumen de cómo interactuar con la pantalla del proyecto, así como también ofrece un breve resumen del proyecto seleccionado. Esta pantalla fue un requerimiento de la plantilla de *analizo.info* y por lo tanto, añadido con posteridad, en las etapas finales del desarrollo.

CU-05: Pantalla de ayuda

Resumen	<i>Esta pantalla se creó a modo de ayuda para el usuario. Se muestra antes de acceder a la pantalla de proyecto y no solo muestra un resumen del mismo, sino también unas instrucciones básicas para familiarizar al usuario con la forma de interactuar con la aplicación.</i>
Importancia	MEDIA – Pese a que esta pantalla no tiene una funcionalidad real, el usuario obtiene información relevante de ella, así como una breve ayuda técnica en caso de necesitarla.
Frecuencia de uso	MEDIA – Esta pantalla se visitará la primera vez que se acceda a un proyecto. Si una vez dentro de este, el usuario desea volver a verla, bastará con que seleccione el botón <i>info</i> dentro de la pantalla de proyecto.
Actores principales	USUARIO
Acciones	<ol style="list-style-type: none"> 1. Desde la pantalla principal, el usuario selecciona un proyecto activo 2. Una vez dentro de la pantalla, se explicará en texto plano la naturaleza del texto seleccionado así como una breve ayuda visual de cómo interactuar con la aplicación 3. Una vez el usuario está satisfecho con la lectura, se accederá a la pantalla del proyecto para iniciar el proceso de análisis
Error	Cualquier error en este elemento será debido a una excepción HW/SW. No hay ningún posible error controlado.
Excepciones	Ninguna contemplada

Tabla 8 - Caso de Uso 5

El siguiente diagrama de casos de uso condensa lo explicado en las anteriores tablas de forma gráfica. Al tratarse de una aplicación relativamente pequeña en términos de desarrollo el diagrama obtenido no es especialmente difícil de comprender.

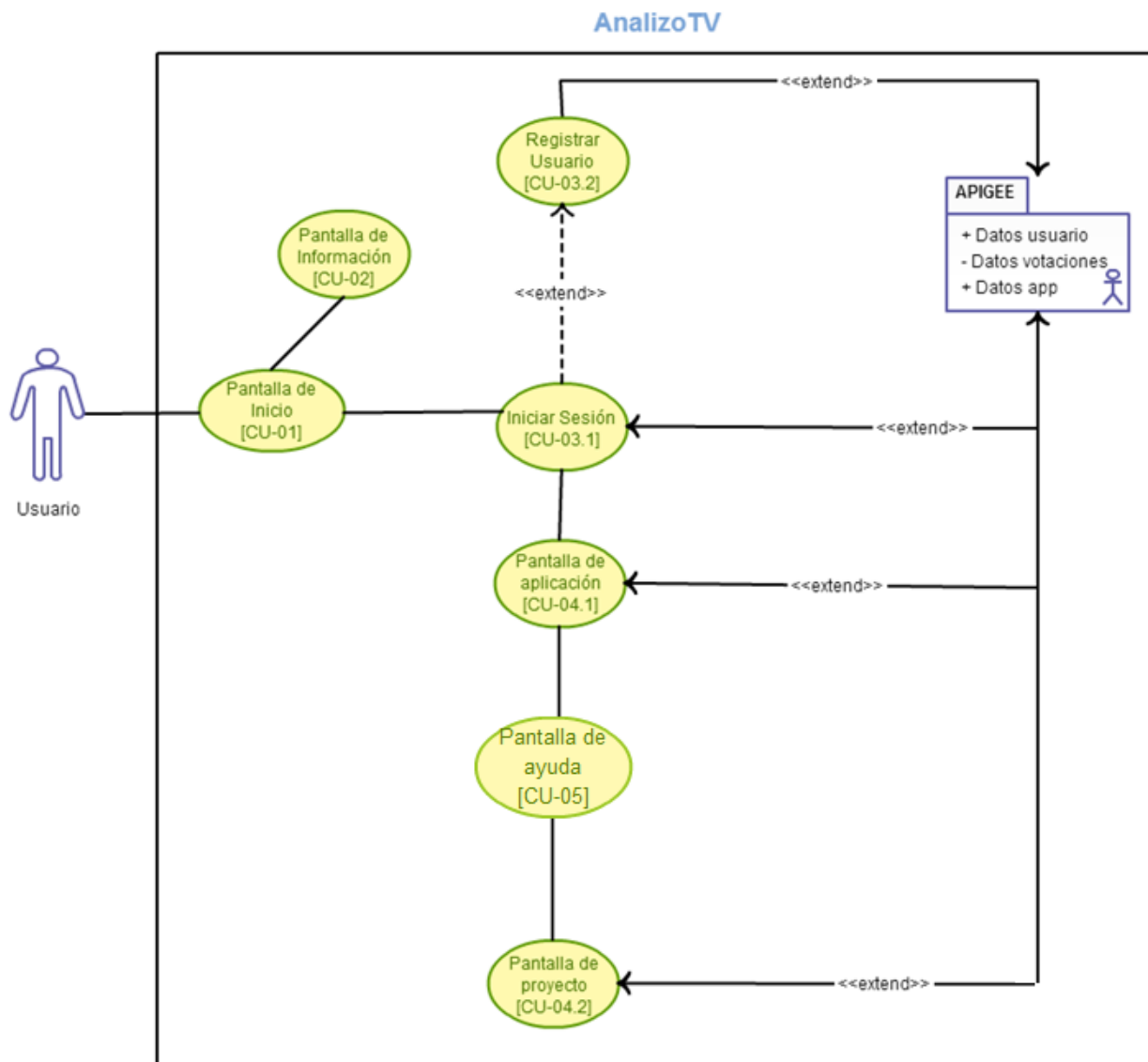


Ilustración 8 - Diagrama Casos de Uso

Mediante los dos diagramas expuestos, así como el desglose por escrito de cada caso de uso, se puede obtener una idea completa de la aplicación final.

Planificación temporal

En todo proyecto es necesario llevar cierto control del tiempo de dedicación a cada etapa del mismo. La filosofía de XP sin embargo, suple esta etapa con constantes reuniones con el cliente.

La filosofía ágil, con sus iteraciones y modificaciones, hace un tanto difícil el planificar una evolución temporal. No obstante, XP anima a los desarrolladores a fijarse objetivos. Estos objetivos a diferencia de otras metodologías no son inflexibles en cuanto al tiempo o al coste, sino más bien susceptibles a cambios.

Con cada iteración y consiguiente reunión con *analizo.info*, se procede a realizar un análisis de los resultados obtenidos hasta el momento, los resultados que se querían obtener en la próxima iteración y el plazo de entrega para la misma.

Esto no implica que no existiera una fecha límite de entrega. Si no se definen unos límites temporales y funcionales, una aplicación es susceptible a no terminarse nunca. En nuestro caso particular nos definimos la meta de tener una versión Beta de la aplicación preparada para Julio como fecha límite, además de fijarnos Septiembre como fecha tope para la entrega de la *release* final junto con esta memoria.

Teniendo en cuenta que el proyecto se empezó en Abril podría dar la impresión de ser unas fechas muy extensas para un proyecto de pequeña envergadura. El hecho es que debido al desconocimiento de la plataforma, la poca documentación encontrada y el reto que suponía abarcar un proyecto real por primera vez, quisimos ser cautos en nuestra estimación temporal. A continuación reflejamos en una tabla la evolución de dicha planificación:

Versión	Cambios	Fecha
0.1	Versión inicial	19/06/2013
0.1.1	Lógica de la aplicación	20/06/2013
0.1.2	Icono. Seguimos trabajando en la lógica	21/06/2013
0.1.3	Obtener información de los canales	22/06/2013
0.1.4	Correcciones. Añadida pantalla splash	22/06/2013
0.1.5	Sincronización y cambio de canal	01/07/2013
0.1.6	Control de tiempos de emisión	03/07/2013
0.2	Versión de pruebas	04/07/2013
0.2.1	Apartado gráfico (Splash)	05/07/2013
0.2.2	Apartado gráfico (Login/Registro)	06/07/2013
0.2.3	Correcciones	07/07/2013
0.2.4	Control del mando (Botones)	10/07/2013
0.3	Versión de prueba completa	14/07/2013
0.3.1	Errores, correcciones, API	18/07/2013
0.3.2	Pantallas About + Descripción	21/07/2013
0.3.3	Corregido error tiempos de emisión	23/07/2013
0.4	Versión final. Listo su lanzamiento	02/08/2013

Tabla 9 - Progreso temporal

Dicha estimación se probó certera, habiendo entregado una Beta final pasado mediados de Julio, dejando las modificaciones finales para la *release* oficial y su entrega a principios de Septiembre. Cabe destacar que tras cada nueva versión se realizaron reuniones para discutir nuevos cambios y mejoras. Dichas reuniones no fueron las únicas, aunque si nos propusimos que fueran obligatorias. Tras cada versión pudimos obtener, junto a *analizo.info*, una idea del progreso y de las impresiones más inmediatas con respecto a la evolución de la aplicación. El desglose detallado de este proceso, con sus tareas e hitos, se describe en los apartados siguientes.

Tecnologías Contempladas

El lector ha de saber que antes de embarcarse en desarrollo de aplicaciones que se ejecutan en televisores y hacen uso de funcionalidades "second-screen", lo ideal y así lo suponemos en este proyecto, sería tener conocimientos previos de las opciones que hay actualmente disponibles en el mercado.

A continuación exponemos las alternativas que contemplamos antes de iniciar este proyecto y cuáles son las razones que nos han llevado a tomar una decisión u otra.

Opciones contempladas

Antes de decantarnos por el entorno que nos ofrece Samsung estuvimos contemplando distintas opciones.

En el mercado de las SmartTV existen diversas tecnologías y fabricantes, cada uno con sus pros y sus contras. A continuación vamos a explicar las alternativas que estuvimos barajando y cuáles fueron las razones que nos llevaron a decantarnos por la televisión de Samsung.

Apple: iOS y Airplay

Apple cuenta en el mercado con un dispositivo conocido como *AppleTV*. Este no soporta aplicaciones desarrolladas por terceros, por el contrario hace uso del protocolo *AirPlay* (anteriormente llamado *AirTunes* cuando solo soportaba audio). Es un protocolo propietario que permite la transmisión inalámbrica de audio, video y fotografías, junto con los metadatos relacionados entre dispositivos. Originalmente sólo se implementó en el software y los dispositivos de *Apple*, sin embargo recientemente *Apple* ha abierto la parte de audio-streaming del protocolo *AirPlay* como una tecnología de componentes para terceros (fabricantes asociados), de forma que puedan utilizarlo en sus productos con el fin de ser compatible con iDevices de *Apple*.

Hay dos tipos de dispositivos *AirPlay*: las que envían los contenidos audiovisuales, y los que son capaces de recibir el contenido y reproducirlo en pantallas y altavoces.

El hardware soportado por *AirPlay* es:

- iMac
- Mac mini
- MacBook Air
- MacBook Pro
- Apple TV (2ª generación y posterior)
- iPhone4S (y posterior)
- iPad2 (y posterior)
- Mini iPad
- iPod Touch (5ª generación y posterior)

El efecto espejo de *AirPlay* es una tecnología ligeramente diferente que permite que el contenido que se emite desde una variedad de dispositivos iOS sea transmitido a una *AppleTV* de segunda generación. En ningún caso este dispositivo es capaz de ejecutar aplicaciones de manera oficial.

Además este sistema, todos los sistemas de Apple Inc. de hecho, son protocolos muy cerrados con un rango de dispositivos muy reducido. Por estos motivos esta opción se descartó desde un principio.

La limitación en su uso y hardware soportado, el coste de las licencias, así como la imposibilidad de disponer de código abierto dieron mucho más peso a otras opciones.

Google TV

GoogleTV, la televisión del gigante Google, al contrario que la competencia, hace uso de *Android*, el sistema móvil por excelencia (con permiso de iOS). Desarrollar para *GoogleTV*, nos pareció muy interesante y estuvimos barajando la posibilidad.

La documentación es excelente y cuenta con la comunidad de Google dispuesta siempre a echar una mano.

La razón por la que la acabamos descartando es por la poca presencia que tiene actualmente en el mercado español. Solo está presente en algunos dispositivos de origen asiático que se conectan al televisor para dotarlo de las capacidades de *SmartTV*. En consecuencia, algunas de las funciones como control por voz o gestos se ven limitadas.

Nos parece que es una opción a tener en cuenta para el mercado americano y quizá en un futuro para el español.

LG SmartTV

LG dispone de televisiones y entorno de desarrollo muy parecidos a Samsung. Estamos hablando de un navegador web que ejecuta tecnologías web en el dispositivo. La diferencia es muy pequeña y al aparecer antes en el mercado, las televisiones de Samsung nos dan más confianza para abordar un proyecto serio.

Si consultamos la web de desarrolladores developer.lge.com, podemos ver, aparte de un **coincidente parecido a la web de Samsung**, una escasa documentación. En los foros hay muchas menos dudas resueltas y las características ofrecidas dan la impresión de no alcanzar a las de su competidor.

Por estas razones no contemplamos LG como una opción.

Samsung SmartTV y AllShare Framework

El SDK de Samsung permite desarrollar aplicaciones de forma muy similar a como desarrollaríamos una página web, basándose en tecnologías como son *HTML5* y *JavaScript*.

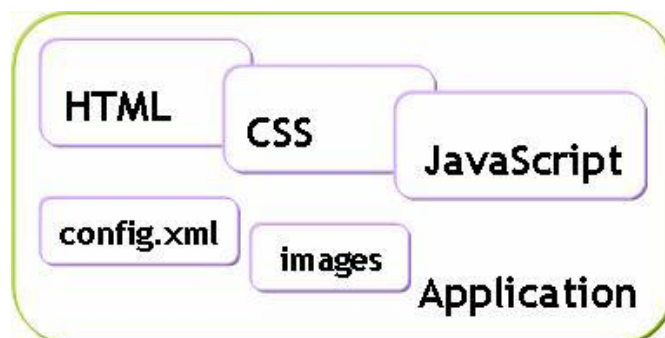


Ilustración 9 - Estructura SDK Samsung

Con HTML5, Samsung ha podido crear un ambiente integrado que soporta el desarrollo de aplicaciones de convergencia. Esto permite que la Smart TV de Samsung interactúe y se comuniquen con dispositivos externos. Además, el SDK 4.0 de *SmartTV* proporciona un ambiente de desarrollo de nube local que permite a los desarrolladores que utilizan OSX o sistemas UNIX trabajar con otros desarrolladores que utilizan Windows.

Si añadimos a esta herramienta el *AllShare Framework*, los desarrolladores disponen de un sistema de fácil integración para la creación de aplicaciones multidispositivo, que permite:

- Control de las Smart TV desde el dispositivo.
- Mirroring de la pantalla del móvil en la TV.
- Hacer streaming de archivos multimedia.
- Compartir archivos con wifi direct.
- Lanzar el browser del PC / TV desde el móvil.

Mediante la tecnología *AllShare*, sistema de conexión de dispositivos, es posible conectar múltiples aparatos con certificación DLNA e interactuar entre ellos de forma cómoda e inalámbrica.

Además *Javascript*, y más concretamente *jQuery*, es un lenguaje muy potente y cada vez más presente en la web. Una enorme cantidad de aplicaciones, tanto móviles como web, están siendo desarrolladas utilizando esta librería que simplifica el uso de una enorme cantidad de funciones. Sin ir más lejos, el nuevo sistema móvil de Mozilla hace uso de esta tecnología de forma nativa para su SO.

Además, el hecho de disponer de una *Samsung SmartTV* hizo que la balanza se decantara a favor de la firma Coreana.

Desarrollo

Versión 0.1 – Nombre *Abigail* (Alpha)

Descripción

Versión inicial de la aplicación. Consta con funcionalidad básica de login contra la web, por el momento forzada, aún no hay interfaz definida. Ya tiene definida estructuras básicas de proyecto y se permite el control de volumen desde la aplicación.

Aún no es funcional ya que no aloja ningún proyecto real ni registra las estadísticas en la web. Falta definir toda la interfaz, así como crear y controlar restricciones.

Problemas conocidos

- No se visualiza imagen, aparece escondida detrás de la capa de la aplicación.
- No se puede cambiar de canal, el usuario está restringido al canal desde el cual se abrió la aplicación

Cambios

commit/01c231f892945116e69812322b3ac8078623316d

A destacar en esta versión

En la pantalla principal tenemos lo siguiente:

```
//Muestra la señal de televisión en el div especificado.
sf.service.PIG.show('tvsignal');

//Hacemos una llamada a la API para obtener todos los proyectos y
los vamos leyendo de uno en uno
var projects = new Usergrid.Collection({ 'client':client,
'type':'projects' });
    projects.fetch(
        function()
        {
            while(projects.hasNextEntity())
            { var project =
projects.getNextEntity();
                $("#lista").append("<li
id="+project.get('name')+ "> Pulsa "+project.get('name')+ " para: "
+ project.get('program') + "</li>"); } //var
            }, //function - EXITO
            function() { alert('read failed');
} //function - FALLO
        ); //fetch
```

En esta misma pantalla, en el control de pulsaciones del mando podemos ver como se obtiene y construye un proyecto concreto de la API:

```
case KEY_1:
    selected_proyect = 1;
    //Creamos el objeto para llamar a la función (Búsqueda por
nombre)
    proyect_load = {
        type:'proyects',
        name:'1'
    };//load

    client.getEntity(proyect_load, function (err, proyect) {
        if (err) { //error }//if
        else {
            //Vaciamos siempre antes de volver a cargar los div
            $("#categorias").empty();
            $("#categorias").append("<li> Pulsa Rojo para: "
+proyect.get('cat1') + "</li>");
            $("#categorias").append("<li> Pulsa Verde para: "
+proyect.get('cat2') + "</li>");
            $("#categorias").append("<li> Pulsa Amarillo para: "
+proyect.get('cat3') + "</li>");
            $("#categorias").append("<li> Pulsa Azul para: "
+proyect.get('cat4') + "</li>");
        }//else
    }//function
    };//getEntity
    break;
```

También podemos destacar para esta versión el funcionamiento de los campos de texto en la SmartTV, así como la función de login en la API:

```
$('#password').sfTextInput({
    text:'',
    maxlength:10,
    oncomplete: function (text)
    {
        if (text) {
            password = text;
            loginUser(); }//if
        else { /*error*/ }//else
    }
});//sfTextInput
$('#user').sfTextInput({
    text:'',
    maxlength:10,
    oncomplete: function (text)
    {
        if (text) {
            username = text;
            $('#password').sfTextInput('focus');
        }//if
        else { /*error*/ }//else
    }//function
});//sfTextInput

function loginUser() {
    client.logout();
```

```

client.login(username, password,
function (err) {
  if (err) {          alert('could not log user in');    }//if
  else {
    alert('user has been logged in');
    client.getLoggedInUser(function(err, data, user)
    {
      if(err) {          alert('could not get logged in
user');          } //if
      else {
        alert('got logged in user');
        sf.scene.hide('Login');
        //Aquí cargamos la escena principal
        sf.scene.show('Default');
        sf.scene.focus('Default');          }//else
      });//getLoggedInUser
    }//else
  });//function
};//loginUser

```

Versión 0.1.1 (Alpha)

Descripción

Avance parcial sobre definición de alpha *release* original 0.1. Se ha añadido función que por el momento es transparente para el usuario, en un futuro controlará la entrada de órdenes a la aplicación.

Se ha corregido el problema de capas entra la aplicación y la imagen de pantalla, ahora se ve la pantalla y la capa de aplicación solo ocupa un pequeño espacio del lateral.

Novedades respecto a versión anterior

- Se ha añadido función para en futuras implementaciones dar funcionalidad a los botones del mando.

Problemas solventados

- Imagen ya se puede visualizar, con marco temporal de aplicación al costado derecho. Es una capa superior a la imagen de la televisión, en un futuro se cambiará su disposición y su control.

Problemas conocidos

- No se puede cambiar de canal, el usuario está restringido al canal desde el cual se abrió la aplicación. Si se intenta cambiar de canal la aplicación se cierra.
- En ocasiones la aplicación no se carga bien debido a que el JavaScript que utiliza necesita unos 2 segundos de margen.

Cambios

[commit/33ac32420cfd36ac2d4f20286c22f2c1faaa6842](https://github.com/33ac32420cfd36ac2d4f20286c22f2c1faaa6842)

A destacar en esta versión

Además del apartado gráfico, en esta versión cabe destacar la nueva forma de cargar las categorías de un proyecto creando una función específica para ello:

```

if (selected_project == -1)
{
    selected_project = 1;
    project_load = {
        type:'projects',
        name:'1'
    };//if

function loadCategories(project_load){

    client.getEntity(project_load, function (err, project) {
        if (err{ /*error*/ }//if
        else {
            $("#categorias").empty();
            $("#categorias").append("<li> Pulsa Rojo para: "
+project.get('cat1') + "</li>");
            $("#categorias").append("<li> Pulsa Verde para: "
+project.get('cat2') + "</li>");
            $("#categorias").append("<li> Pulsa Amarillo para: "
+project.get('cat3') + "</li>");
            $("#categorias").append("<li> Pulsa Azul para: "
+project.get('cat4') + "</li>");
            $("#lista").hide();
            $("#categorias").show();
        }//else
    });//getEntity
};//loadCategories

```

Versión 0.1.2 (Alpha)

Descripción

Sobre los cambios de las alpha *release* 0.1.2 se ha añadido un valor visual en forma de icono de la aplicación. Se ha creado y añadido al proyecto una imagen que actúa a forma de icono una vez instalada en el dispositivo.

En fases posteriores del proyecto, cuando se desarrolle el aspecto visual se integrará a la estética del mismo.

Novedades respecto a versión anterior

- Se añade icono específico para la aplicación. Al instalarla ya no se le asigna un icono genérico, sino el diseñado y especificado por nosotros.

Problemas solventados

-

Problemas conocidos

- No se puede cambiar de canal, el usuario está restringido al canal desde el cual se abrió la aplicación. Si se intenta cambiar de canal la aplicación se cierra.
- En ocasiones la aplicación no se carga bien debido a que el JavaScript que utiliza necesita unos 2 segundos de margen.

Cambios

[commit/17964bf093cd74a47497393e090e6b877f15b600](https://github.com/commit/17964bf093cd74a47497393e090e6b877f15b600)

A destacar en esta versión

Al haber únicamente añadido el logotipo no hay ninguna parte del código digna de destacar en el documento.

Versión 0.1.3 (Alpha)

Descripción

Se añade y amplía la funcionalidad disponible con respecto a la alpha *release* 0.1.2. De cara a la funcionalidad final de la aplicación se añade función para obtener información de los canales, como puede ser nombre, sintonía, banda...

Esta información en futuras implementaciones nos permitirá controlar la restricción de los canales al trabajar con proyectos específicos.

Novedades respecto a versión anterior

- Función para detectar y almacenar información de los canales sintonizados

Problemas solventados

- Se puede cambiar de canal mientras se está dentro de la aplicación, ya no se está restringido al canal desde el que se abrió la misma. Este cambio es temporal, todavía no se ha decidido si se mantendrá de algún modo o se desechará una vez se añada la restricción temporal y de canal respecto a los proyectos.

Problemas conocidos

- En ocasiones la aplicación no se inicia bien debido a que el JavaScript que utiliza necesita unos 2 segundos de margen para terminar la carga.

Cambios

[commit/2ccbcf5bc31efa0489833d67c924935e3b6a4533](https://github.com/2ccbcf5bc31efa0489833d67c924935e3b6a4533)

A destacar en esta versión

En esta versión se ha añadido un método para obtener la información de los canales y del programa para aprovecharlo en un futuro:

```

channel = webapis.tv.channel.getCurrentChannel();
program = webapis.tv.channel.getCurrentProgram();

var channelInfo = {
    name: channel.channelName,
    ptc: channel.ptc,
    major: channel.major,
    minor: channel.minor,
    lcn: channel.lcn,
    sourceID: channel.sourceID,
    programNumber: channel.programNumber,
    transportStreamID: channel.transportStreamID,
    originalNetworkID: channel.originalNetworkID,
    serviceName: channel.serviceName,
    type: 'channelinfos'
}; //channelInfo

client.createEntity(channelInfo, function (err,
channelInfo) {
    if (err) {
        alert('entity not
created');
    } //if
    else {
        alert('entity created');

        channelInfo.save(function(err){
            if (err) { /*error*/ } //if
            else { /*success*/ } //else
        }); //save
    } //else
}); //function

```

Versión 0.1.4 (Alpha)

Descripción

En esta *release* se corrige el problema conocido desde la *release* original *Abigail*. Debido a problemas de carga en ciertas funciones JavaScript utilizadas en el proyecto, tras la primera carga hay problemas para cargar correctamente la aplicación ya que no hay tiempo suficiente para ejecutar el Javascript.

Damos solución a esto creando una pantalla de splash temporal, appena se visualiza ni tiene diseño, simplemente implementamos una espera de 5 segundos antes de arrancar funciones principales para que Javascript tenga tiempo de cargarse.

Además de esta forma conseguimos que si la sesión del usuario a expirado este vuelva a ser llevado a la pantalla de login en vez de a la principal.

Novedades respecto a versión anterior

- Ventana de splash con duración 5 segundos para dar tiempo a funciones y librerías externas a cargar de forma correcta

Problemas solventados

- La aplicación carga correctamente tras introducir breve pantalla de carga. JavaScript carga correctamente con apenas 2 segundos de espera

Problemas conocidos

-

Cambios

commit/bd6b5922ac1bd5449b61605fb8150c68a5f2dd4a

A destacar en esta versión

A continuación se muestran los cambios en el código específicos a la pantalla de splash:

```
$('#loading').sfLoading('show');

setTimeout(
  function(){
    if (client.isLoggedIn()) {
      client.getLoggedInUser(function(err, data, user) {
        if(err) {
          alert('could not get logged in user');
          client.logout();
          $('#loading').sfLoading('hide');
          sf.scene.hide('Splash');;
          sf.scene.show('Login');
          sf.scene.focus('Login');

        } //if
          else {    alert('got logged in user:
'+user.get('username')); }//else
          );//getLoggedIn

          $('#loading').sfLoading('hide');
          sf.scene.hide('Splash');
          sf.scene.show('Default');
          sf.scene.focus('Default'); }//if
        else{
          $('#loading').sfLoading('hide');
          sf.scene.hide('Splash');
          sf.scene.show('Login');
          sf.scene.focus('Login');      }/*else*/},
5000);//setTimeout
```


Versión 0.1.5 (Alpha)

Descripción

Haciendo uso de funciones implementadas en releases anteriores, se añade funcionalidad para sintonización de canales. Tras esta *release* se podrá restringir el canal una vez seleccionado el proyecto.

Cuando un proyecto se selecciona, automáticamente se cambia al canal específico al mismo, haciendo uso del nombre del canal que obtenemos de la información que conseguimos del mismo a través de la función implementada en *release* 0.1.3.

Novedades respecto a versión anterior

- Al seleccionar un proyecto se cambia y sintoniza automáticamente al canal sobre cual trata el mismo

Problemas solventados

-

Problemas conocidos

- Al añadir la sintonización de canales, surgen conflictos con la posibilidad de cambiar canal. Una vez se ha seleccionado el proyecto, el usuario no debería poder cambiar de canal, para asegurar fiabilidad de la votación.

Cambios

commit/93327adbae07dad623275176d98c8737acccae60

A destacar en esta versión

En esta versión nos hemos dado cuenta que la información obtenida de los canales en anteriores versiones no nos es útil debido a que cada televisión tiene sus propios datos de sintonización de canales. Para resolver este conflicto lo hicimos buscando el nombre del canal en la lista de canales disponible en la televisión (el nombre del canal es emitido por la cadena y por tanto es único), y de esta forma leemos los datos del canal y procedemos a sintonizar de la siguiente forma:

```
function tuneChannel(proyect_load){
var MyChannel;

client.getEntity(proyect_load, function (err, proyect) {
  if (err) { /*error*/ } //if
  else {
    try {
      // gets 100 channel list of Digital channels
      webapis.tv.channel.getChannelList(function
(TVChannels) {
        for(i=0;i<TVChannels.length;i++)
          {
            if(proyect.get('OnChannel') ==
TVChannels[i].channelName)
              { MyChannel=TVChannels[i]; } //if

          } //for } /*function*/, errorCallback,
webapis.tv.channel.NAVIGATOR_MODE_DIGITAL, 0,
```

```

100); //getChannelList

        webapis.tv.channel.tune({
            ptc: MyChannel.ptc,
            major: MyChannel.major,
            minor: MyChannel.minor,
            sourceID: MyChannel.sourceID,
            programNumber: MyChannel.programNumber,
            transportStreamID: MyChannel.transportStreamID,
            tunecallback: {
                onsuccess: function(programList) {
                    console.log("getting program list is successfully");
                } /*function*/,
                onerror: function(channelList) {
                    console.log("getting program list is successfully");
                } /*function*/,
            } //tunecallback
        }, successCB, errorCB, 0); //tune

        alert('Channel tuned');
    } //try
        catch (error) { alert(error); } //catch

    } //else
}); //getEntity
} //tuneChannel

```

Versión 0.1.6 (Alpha)

Descripción

A fin de ampliar la funcionalidad de la anterior release, se añade función para controlar el tiempo de emisión de programas. Con esta novedad, una vez el usuario ha seleccionado un proyecto, la aplicación comprueba que el usuario se encuentra dentro de la franja horaria adecuada. De no ser así, la aplicación no registrará ni permitirá que se envíe el voto.

Novedades respecto a versión anterior

- Restricción temporal con respecto al proyecto, canal y emisión relacionada con el mismo. De esta forma, junto con restricción de canal, se afianza la validez del voto.

Problemas solventados

-

Problemas conocidos

- Al añadir la restricción temporal y de sintonía, surgen conflictos con la posibilidad de cambiar canal. Una vez se ha seleccionado el proyecto, el usuario no debería poder cambiar de canal ni programa, para asegurar fiabilidad de la votación.

Cambios

[commit/04649e3441cbc28ad71f3ea187bacd0b0dd65aee](https://github.com/commit/04649e3441cbc28ad71f3ea187bacd0b0dd65aee)

A destacar en esta versión

En esta versión se pretende controlar el tiempo de emisión de un programa para que solo esté disponible en ese periodo de tiempo. En primera instancia se introdujo un grave error en el momento de comprobar cuál es la hora de finalización de la emisión. Para evitar confusiones dejaremos la versión corregida a continuación:

```
function loadProgram(proyect_load){
    now=new Date();
    var weekday=new Array(7);
    weekday[0]="Sunday";
    weekday[1]="Monday";
    weekday[2]="Tuesday";
    weekday[3]="Wednesday";
    weekday[4]="Thursday";
    weekday[5]="Friday";
    weekday[6]="Saturday";

    var now_day = weekday[now.getDay()];

    client.getEntity(proyect_load, function(err, proyecto){
        if (err) { /*error*/ } //if
        else {
            var proyect_day = proyecto.get(now_day);

            if(proyect_day){
                var hora = '';
                var inicio_p = '';
                var fin_p = '';
                //Leemos la haciéndola universal en formato HHMM 1030 siempre será
                //menor que 1100
                //Si los minutos son menores que 10 le añadimos un 0 delante para
                //evitar problemas
                if (now.getMinutes() < 10)
                    { hora =
''+now.getHours()+ '0'+now.getMinutes(); } //if
                else
                    { hora =
''+now.getHours()+ ''+now.getMinutes(); } //else

                inicio_p
                =' '+proyecto.get('HoraInicio')+ ''+proyecto.get('MinutoInicio');
                fin_p =
''+proyecto.get('HoraFin')+ ''+proyecto.get('MinutoFin');

                //Nos aseguramos que Javascript lee la variable como un entero
                if(parseInt(hora) >= parseInt(inicio_p))
                {
                    if(parseInt(hora) <= parseInt(fin_p))
                    {
                        loadCategories(proyect_load);
                        tuneChannel(proyect_load);

                    } //if fin_p
                } else{
selected_project =-1; } //else fin_p
```

```
        } if inicio_p
        else{
selected_proyect =-1; }//else inicio_p
        } //if proyect_day
        else{ selected_proyect =-1;
}//else proyect_day
        }//else
    });//getEntity
}//loadProgram
```

Versión 0.2 – Nombre *Bibi* (Alpha)

Descripción

Con las anteriores versiones cerradas y las principales funciones implementadas, se crea la *release Bibi* como versión preliminar para su evaluación por los beta-testers seleccionados para este proyecto.

Con toda la funcionalidad hasta el momento afianzada, se le da lógica a la aplicación, añadiendo proyectos y ejemplos creíbles aunque todavía no reales.

Novedades respecto a versión anterior

- Se da funcionalidad a los botones de numeración y colores del mando, estos son los encargados de seleccionar el proyecto y elegir temática de votación, respectivamente.

Problemas solventados

-

Problemas conocidos

- Al añadir la restricción temporal y de sintonía, surgen conflictos con la posibilidad de cambiar canal. Una vez se ha seleccionado el proyecto, el usuario no debería poder cambiar de canal ni programa, para asegurar fiabilidad de la votación.

A destacar en esta versión

Al haber únicamente añadido más opciones a las ya existente no hay parte del código digna de destacar en el documento, el código añadido es una réplica variante a la ya existente.

Junta de revisión 03/07/2013

Una vez terminados las funcionalidades básicas nos reunimos con analizo.info para que nos dieran sus primeras impresiones. En esta reunión sobre todo tratamos como sería la estética de la aplicación y como se presentaría la información al usuario.

Versión 0.2.1

Descripción

A fin de mejorar el aspecto visual de la aplicación, se ha editado la pantalla de carga inicial de la que consta la aplicación, pantalla que en su momento fue temporal en *release* 0.1.4. Ahora se ha añadido una animación de carga, así como texto para que sea más intuitiva de cara al usuario.

Novedades respecto a versión anterior

- Pantalla de splash animada y editada para que actúe de forma definitiva como pantalla de carga.

Problemas solventados

-

Problemas conocidos

- Al añadir la restricción temporal y de sintonía, surgen conflictos con la posibilidad de cambiar canal. Una vez se ha seleccionado el proyecto, el usuario no debería poder cambiar de canal ni programa, para asegurar fiabilidad de la votación.
- A la hora de utilizar botones especiales del mando, como pueden ser “Herramientas”, “Info”,...la aplicación parece no detectar selección. Es posible que allá que utilizar nuevas funciones debido a la naturaleza de los mismos.

Cambios

<commit/f651b1ba577b7b7e4061813ace6d3fe6b93567d0>

A destacar en esta versión

Al haber únicamente modificado la pantalla de splash no hay parte del código digna de destacar en el documento. Los cambios efectuados son más notables en el aspecto estético, no tienen gran impacto sobre el código.

Versión 0.2.2

Descripción

En esta release, a partir de los cambios ya implementados, se edita la pantalla de login/registro con la estética visual acorde a analizo.info.

Novedades respecto a versión anterior

- Se edita la pantalla de login y registro, que controla identidad de usuarios.

Problemas solventados

-

Problemas conocidos

- Si usuario no está registrado la aplicación no cambia, se mantiene en la misma pantalla y no muestra error.
- A la hora de utilizar botones especiales del mando, como pueden ser “Herramientas”, “Info”,...la aplicación parece no detectar selección. Es posible que allá que utilizar nuevas funciones debido a la naturaleza de los mismos.

Cambios

<commit/242551a8711a9c3a486a3fc288f356cf812e51b2>

A destacar en esta versión

Al igual que la *release* anterior, los cambio en esta ocasión son puramente estéticos. La funcionalidad sigue siendo la misma, no hay parte del código digna de destacar en el documento

Junta de revisión 08/07/2013

Seguimos tratando como se debería comportar la interfaz y corregimos pequeños errores.

Problemas y/o mejoras detectadas:

- Quitar el "de" de un proyecto de analizo.info
- Subir un poco el teclado para que no se corte con el margen inferior
- Mostrar un OK cuando el registro esté completo
- El botón izq en la pantalla de login falla
- Botón return no funciona
- Mostrar, en estos momentos no hay ningún proyecto si es preciso
- Generar la lista de forma dinámica mostrando los que están en emisión y pueden seleccionarse y el resto a la hora que empiezan

Versión 0.2.3 / 0.2.4

Descripción

Se corrigen la longitud máxima de los campos y se controlan las teclas especiales del mando para poder usarlas como acciones.

Novedades respecto a versión anterior

- Tenemos acceso a las teclas especiales del mando con lo que podemos introducir acciones como volver/salir/desconectarse y bloquear el cambio de canal.

Problemas solventados

- Al añadir la restricción temporal y de sintonía, surgen conflictos con la posibilidad de cambiar canal. Una vez se ha seleccionado el proyecto, el usuario no debería poder cambiar de canal ni programa, para asegurar fiabilidad de la votación.
- A la hora de utilizar botones especiales del mando, como pueden ser "Herramientas", "Info",...la aplicación parece no detectar selección. Es posible que allá que utilizar nuevas funciones debido a la naturaleza de los mismos.

Problemas conocidos

- Se sigue sin capturar errores en el caso de login y registro. Los errores típicos como pueden ser: campos vacíos, usuario desconocido o contraseñas incorrectas no están filtrados. La aplicación no arroja ningún error ni da ninguna indicación.

Cambios

[commit/0c0469c9a7682e278f243dbd7f343ff4f7ab3182](#)

[commit/80479448db18dec534e350617bf76cce8037a327](#)

A destacar en esta versión

Para poder utilizar los botones de funcionalidad especiales se hace de la siguiente forma:

```
//Registramos teclas especiales que por defecto las aplicaciones
no tienen en cuenta
sf.key.registerKey(sf.key.CH_UP);
sf.key.registerKey(sf.key.CH_DOWN);

//si pulsamos la tecla deseada
case sf.key.EXIT:
    //prevenimos su comportamiento por defecto
    event.preventDefault();
    //incorporamos el nuevo comportamiento, en este caso salir de
la aplicación
    sf.core.exit(false);
    break;
```


Versión 0.3 – Nombre *Cook* (Alpha)

Descripción

Primera versión completa con todas las interfaces listas. Añadimos la función `doStart()` en la pantalla Default para iniciar todos los elementos gráficos.

Novedades respecto a versión anterior

- Todos los elementos gráficos están listos.

Problemas solventados

- Muchos de los problemas de la última junta de revisión corregidos

Problemas conocidos

- Se sigue sin capturar errores en el caso de login y registro. Los errores típicos como pueden ser: campos vacíos, usuario desconocido o contraseñas incorrectas no están filtrados. La aplicación no arroja ningún error ni da ninguna indicación.
- Aún falta incorporar algunos elementos como son la barra de ayuda.
- Mostrar, en estos momentos no hay ningún proyecto si es preciso

Cambios

commit/cd71c9f9b5ffa5ad924a4d5c313b6d05be03de69

A destacar en esta versión

Esta versión condensa todos los cambios efectuados con anterioridad, así como introduce ciertas mejoras en el plano estético. No hay ningún fragmento de código digno de destacar que no se haya expuesto ya.

Junta de revisión 15/07/2013

Con la interfaz prácticamente terminada podemos ver como se comporta un usuario real y los problemas que tiene de interacción con la televisión a través del mando y las distintas pantallas.

Algunos de los cambios propuestos se recogen a continuación.

Problemas y/o mejoras detectadas

- Cambiar colores logo (no se corresponden con los botones del mando)
- Cambiar momento en que se muestra la alerta de CheckedIn (no esperar a que la API termine)
- Añadir no hay programas en emision/siguientes si es preciso
- Añadir A B C D en las cajas de colores en la pantalla de categorías
- Al hacer logout no se recarga a señal de televisión
- Añadir texto utilice botones ABCD en la pantalla de categorías
- Manejar errores
- Mismo ancho para la tabla de próximamente
- Poner de otro color la barra dónde se muestra la hora de los próximos programas

- para evitar confusiones
- Sombrear los números de los programas en emisión para no confundirse con la imagen de fondo
- Añadir una descripción en pantalla categorías para guiar al usuario
- Añadir barra inferior
- Nueva pantalla about sobre analizo.info

Versión 0.3.1

Descripción

Se corrigen gran parte de los problemas detectados en la última junta y se introduce la versión final del diseño de la API para guardar los CheckIn de cada proyecto. (Se tratará detalladamente más adelante). También se manejan los errores y se avisa al usuario en caso de Registro satisfactorio/erróneo.

Novedades respecto a versión anterior

- Versión final del comportamiento de la API
- Muchas mejoras a nivel de uso
- Manejo de errores

Problemas solventados

- Muchos de los problemas de la última junta de revisión corregidos

Problemas conocidos

- Añadir barra inferior
- Falta la nueva pantalla about

Cambios

[commit/2880db0bec29ad5c9845884988bd7c7961681d12](#)
[commit/32cdad877668ed4cbcc7fb2bf3125da72defc0a9](#)
[commit/dccf76df079abe98b4cfdfe9bfd79fde81bab2](#)

A destacar en esta versión

Esta versión condensa todos los cambios efectuados con anterioridad, así como introduce ciertas mejoras en el plano estético. No hay ningún fragmento de código digno de destacar que no se haya expuesto ya, esta revisión y sucesivas aspiran a modificar o alterar aspectos ya implementados.

Junta de revisión 18/07/2013

Seguimos mejorando la experiencia de usuario y corrigiendo pequeños errores. A estas alturas del proyecto las mejoras y cambios estéticos toman un primer plano, mientras que los errores y pequeñas correcciones empiezan a ser menos frecuentes.

Problemas y/o mejoras detectadas

- Corregir descripción (vaciar lista para evitar concatenamiento)
- Corregir error login (aparece cuadro rojo)
- Cuando acabe la hora de emisión cerrar análisis
- Añadir pantalla de descripción sobre el proyecto a analizar
- Cambiar la descripción en la barra superior por la pregunta
- Mantener logo de analisis.info arriba a la derecha mientras se analiza el programa
- Guardar segundo del checkin en la api
- Cambiar Programas en emisión por Ahora puedes analizar:
- Bajar la alerta de checkin en la posición de la pantalla

Versión 0.3.2

Descripción

En esta versión se incorpora la pantalla de About y la pantalla de descripción del proyecto antes de empezar a analizar. También se corrigen los errores de la última junta y algunos que quedaban pendientes.

Novedades respecto a versión anterior

- Nueva pantalla de información
- Nueva pantalla antes de empezar a analizar un proyecto
- Muchas mejoras a nivel de uso

Problemas solventados

- Muchos de los problemas de la última junta de revisión corregidos

Problemas conocidos

-

Cambios

[commit/8e40d3883352c6994c906713edd3659dcccef85a](#)

[commit/f0cc1c4404727eac50cfa16c044e67a6d32911d3](#)

[commit/0edc16894017961d33938e0dba055181c63edf94](#)

A destacar en esta versión

No hay ningún fragmento de código digno de destacar que no se haya expuesto ya, esta revisión y sucesivas aspiran a modificar o alterar aspectos ya implementados.

Junta de revisión 22/07/2013

Seguimos mejorando la experiencia de usuario corrigiendo errores en la interfaz y profundizando en la pantalla de información de los proyectos para que el usuario no se sienta perdido y sepa que hacer en todo momento.

En este momento nos dimos cuenta del error introducido a la hora de controlar los tiempos de emisión como ya se ha comentado anteriormente.

Problemas y/o mejoras detectadas

- Cambiar Analistas en la comunidad en la barra de ayuda
- Añadir pulsa ok para introducir dato/Iniciar sesión/Comenzar...
- En la pantalla de la información del proyecto apoyar con imágenes en las descripciones de las categorías
- Que aparezcan los botones ABCD alternándose con las imágenes a la hora de mostrar la información
- Gran error a la hora de controlar los tiempos de emisión

Versión 0.3.3

Descripción

Esta versión se basa simplemente en las correcciones de la última junta, corrigiendo así el error en los tiempos de emisión

Novedades respecto a versión anterior

- Correcciones a nivel gráfico y de interfaz
- Solventado gran error en el control de los tiempos de emisión.

Problemas solventados

- Muchos de los problemas de la última junta de revisión corregidos

Problemas conocidos

-

Cambios

[commit/c61c7b6ba0f77b7d8e2ecb41b25cd0c4b7a4b15d](#)
[commit/50f57e5888c868b46d96696dd7624b14c58f5d23](#)
[commit/a65ca5d6724334e0b15ebdf9acefe475f866c2d2](#)
[commit/8a6578dc5949d9c351262af9da9f6a9141a689b6](#)

A destacar en esta versión

No hay ningún fragmento de código digno de destacar que no se haya expuesto ya, esta revisión y sucesivas aspiran a modificar o alterar aspectos ya implementados.

Junta de revisión 24/07/2013

En esta última reunión hablamos de cómo intentar mejorar la pantalla de información de los proyectos y pequeños cambios estéticos para mejorar la usabilidad de la aplicación.

También hablamos de futuras mejoras de la aplicación que serán tratadas más tarde en este documento.

Problemas y/o mejoras detectadas

- Correcciones a nivel visual
- Posibles mejoras

Versión 0.4 - Effy (Beta)

Descripción

Pequeñas correcciones visuales. Primera beta apta para ser utilizada por los usuarios de analizo.info

Novedades respecto a versión anterior

- Correcciones a nivel gráfico y de interfaz

Problemas solventados

- Muchos de los problemas de la última junta de revisión corregidos

Problemas conocidos

-

Cambios

commit/b22412947889a43e418d5e1f345e0760d559de24

Resultados y posibles mejoras

Resultado final

Vamos a hacer un recorrido por el resultado final de la aplicación de manera que de una manera muy visual queden claras todas las funcionalidades de la aplicación.

En cada pantalla procederemos con una breve descripción, cuáles han sido las limitaciones y mejoras que podrían hacerse en futuras versiones.

Pantalla de Splash



Ilustración 10 - Pantalla de Splash

Pantalla inicial que se muestra durante los primeros 5 segundos a fin de que las librerías de la aplicación se cargue correctamente en un segundo plano.

Funcionalidad ampliada y explicada en CU-01: Pantalla de Inicio.

Limitaciones

En el servicio que utilizamos a modo de API (Apigee) hay un error conocido en el que pese a especificar que la sesión de usuario no caduque, esta lo hace a las 24h.

Mejoras

Cuando Apigee corrija este problema, AnalizoTV debería a empezar a mantener la sesión abierta de forma ilimitada.

Pantalla de Login

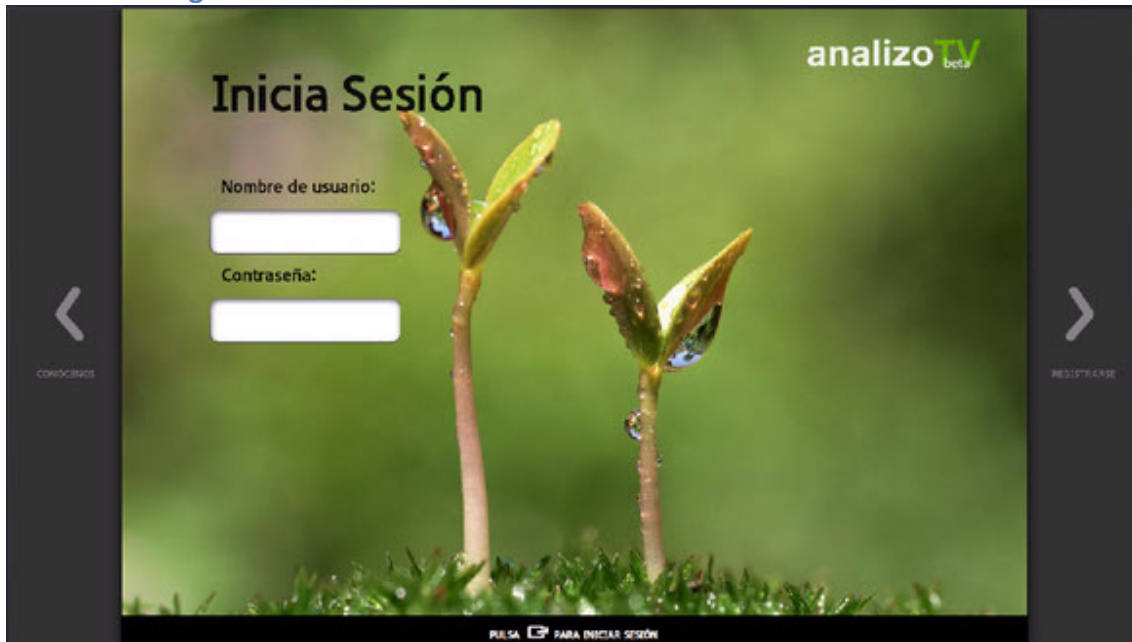


Ilustración 11 - Pantalla de login

Pantalla de login, todo usuario deberá pasar por ella para acceder a la aplicación. Su principal función es la de autenticar al usuario.

Funcionalidad ampliada y explicada en CU-03.1: Pantalla de login.

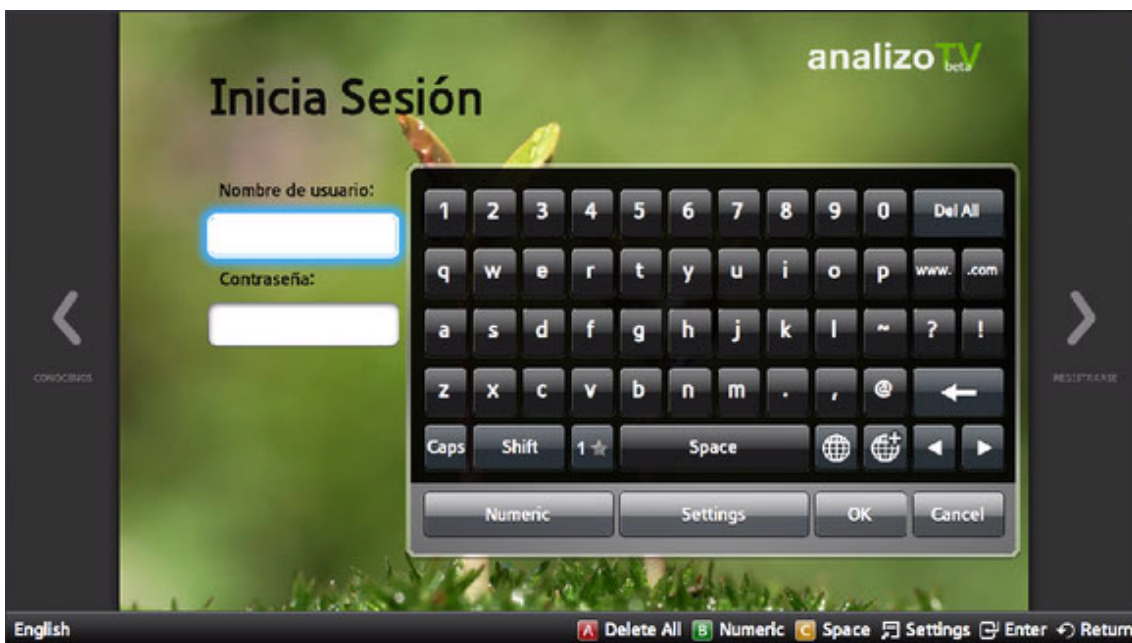


Ilustración 12 - Pantalla de login

Limitaciones

El SDK de Samsung no posee actualmente un input para introducir contraseñas. Además, no existe una forma sencilla de vaciar los input al una vez acabado el proceso de inicio de sesión, por lo que si el usuario se desconecta en la misma sesión, aparecerá el nombre de usuario y contraseña en la pantalla.

Mejoras

Corregir el input de contraseñas cuando Samsung proporcione uno. Conseguir limpiar de alguna forma los formularios de la aplicación.

También se podría mejorar la experiencia de usuario si añadiéramos una visualización de los proyectos disponibles actualmente de forma que el usuario obtenga más información y se decida a hacerse una cuenta y empezar a analizar contenidos.

Pantalla de registro

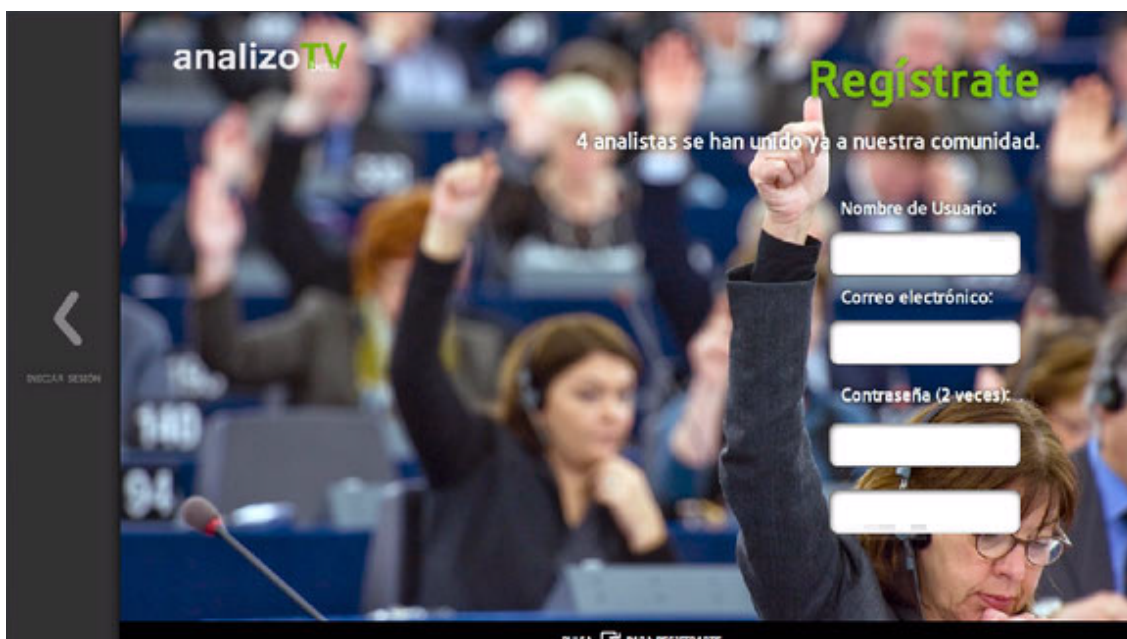


Ilustración 13 - Pantalla de registro

Pantalla cuya finalidad es la de facilitar datos de acceso a nuevos usuarios sin datos de registro anteriores.

Funcionalidad ampliada y explicada en CU-03.2: Pantalla de registro.

Limitaciones

Tiene las mismas limitaciones que la pantalla anterior.

Mejoras

Una posible mejora de esta pantalla sería comprobar que el correo electrónico es un correo válido y lanzar un mail de confirmación hacia el usuario. De momento Apigee no ofrece esta posibilidad.

Pantalla de información



Ilustración 14 - Pantalla de información

Pantalla con la finalidad de ambientar y familiarizar al usuario con la política y principios de *analizo.info*. La pantalla tiene una finalidad meramente informativa.

Funcionalidad ampliada y explicada en CU-02: Pantalla de información.

Pantalla Principal



Ilustración 15 - Pantalla principal

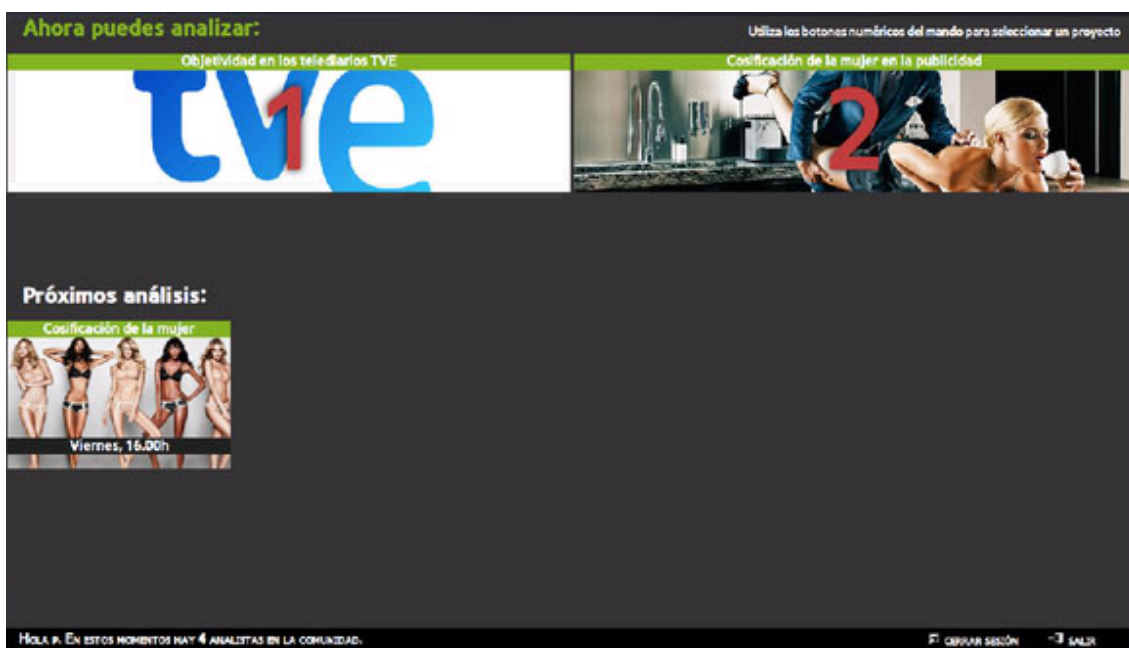


Ilustración 16 - Pantalla principal

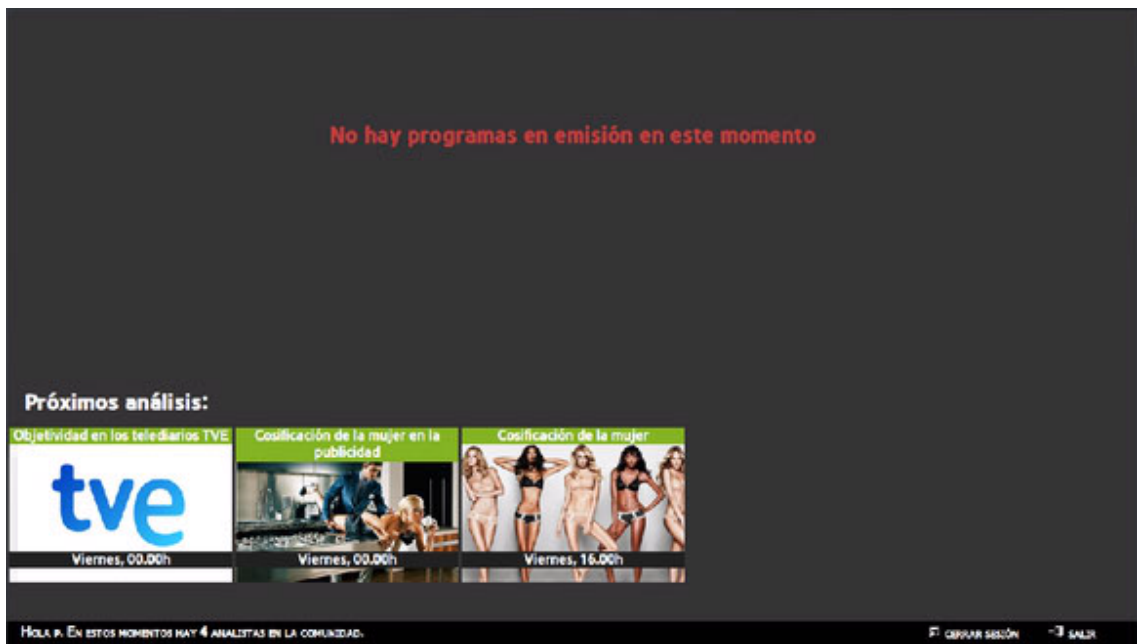


Ilustración 17 - Pantalla principal

Pantalla cuya principal función es la de mostrar los proyectos disponibles. Aquellos proyectos a los que puedan accederse de forma inmediata aparecerán en la parte superior de la aplicación. Aquellos que por el contrario no están disponible por estar fuera de su franja horario aparecerán en la parte inferior.

Funcionalidad ampliada y explicada en CU-04.1: Pantalla de aplicación.

Limitaciones

La librería jQuery que utilizan las SmartTV de Samsung está un poco obsoleta, por lo que no ofrece la posibilidad de hacer llamadas push y pull que podrían ser muy útiles en esta pantalla.

Mejoras

Con las llamadas mencionadas anteriormente podríamos hacer un refresco de los programas en emisión sin que supusiera una carga muy pesada para la aplicación. De esta forma no haría falta salir de la aplicación para que se recarguen los programas como pasa actualmente.

Otra mejora a nivel de Usuario podría ser la de ordenar lo próximos programas por orden de emisión.

Pantalla de información de proyecto

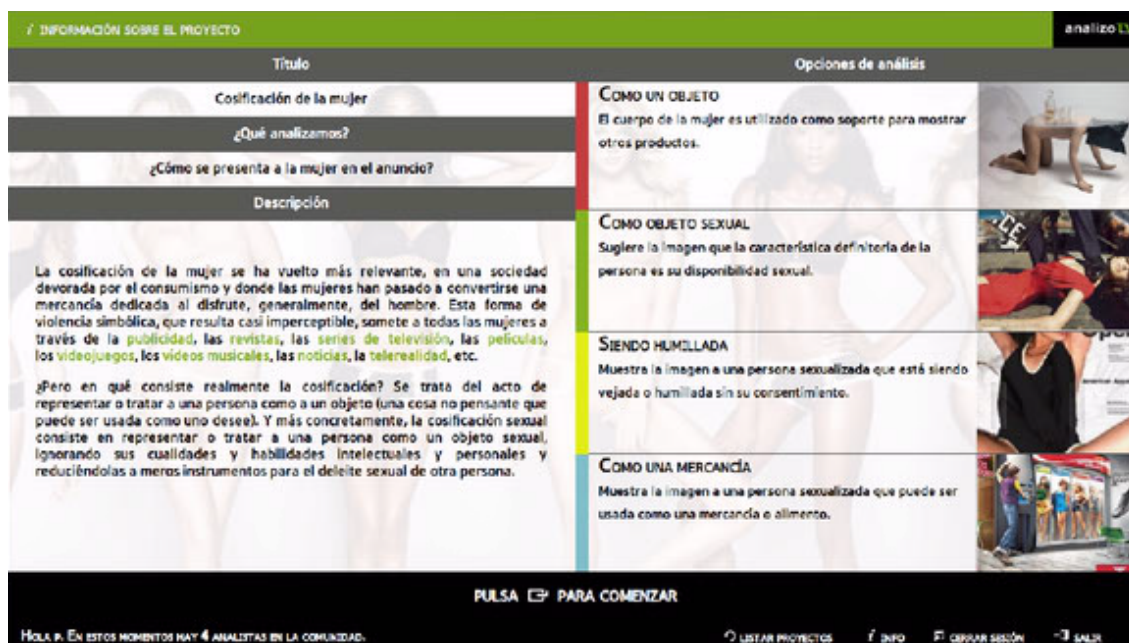


Ilustración 18 - Pantalla de información de proyecto

A modo de ayuda de cara al usuario se ha añadido una pantalla con una breve descripción de cada proyecto, así como instrucciones básicas de como interactuar con la aplicación.

Pantalla de proyecto

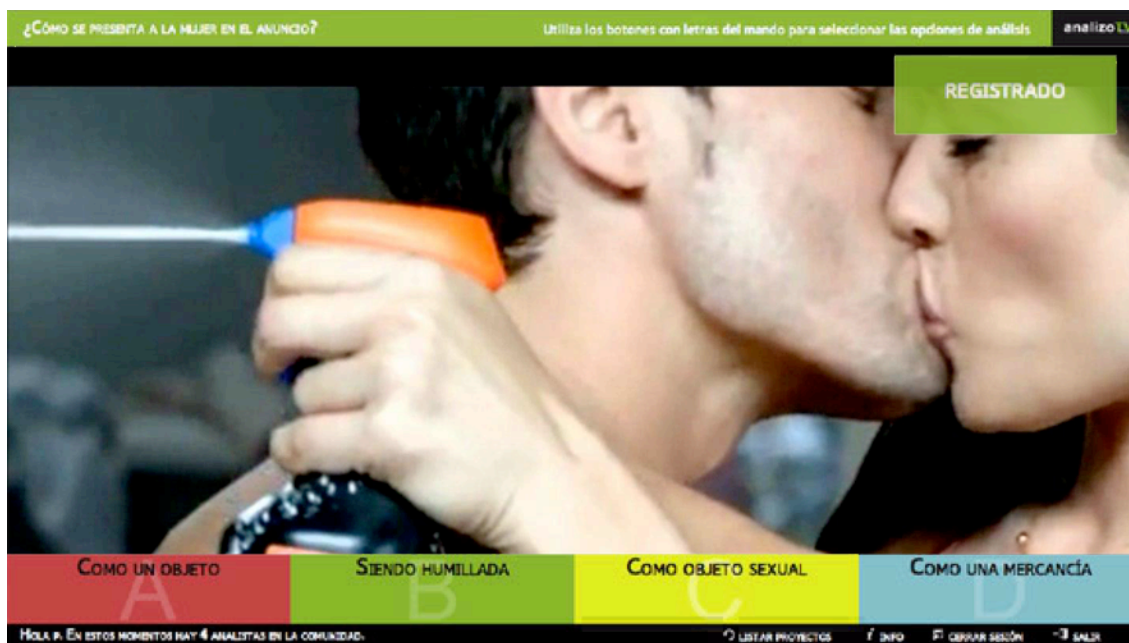


Ilustración 19 - Pantalla de proyecto

Pantalla que muestra los distintos votos disponibles dentro de un proyecto en cuestión. Una vez se seleccione una de ellas el voto será enviado y registrado.

Funcionalidad ampliada y explicada en CU-04.2: Pantalla de proyecto.

Limitaciones

Actualmente esta aplicación no posee limitaciones

Mejoras

Nuevo para mejorar la experiencia de uso, en futuras versiones, se podría habilitar una opción para que el texto del análisis actual pudiera posicionarse arriba, izquierda o derecha según prefiera el usuario.

Otras mejoras

Además de las mejoras previamente vistas, pensamos que la carga de la aplicación se podría optimizar reduciendo el número de llamadas que se repiten contra la API, haciendo una sola llamada a la API en un momento determinado y guardando la información en local e ir actualizándola en segundo plano.

También podría resultar interesante hacer los proyectos más versátiles para distintos tipos de análisis de manera que se pudiera definir un proyecto:

- En varios canales (para analizar pausas publicitarias por ejemplo).
- Habilitar el cambio de canal al usuario si el proyecto lo requiere.
- Definir múltiples horarios.

Otra posible mejora sería aunar los usuarios de la aplicación con los actuales analistas de analizo.info. Debido a la forma en que está construido analizo.info y las limitaciones que tiene javascript al interactuar con bases de datos clásicas esto no es posible tratarlo en este proyecto, pero es un punto a tener en cuenta en un futuro.

En cuanto a limitaciones, la gran limitación de esta aplicación son las políticas de Samsung en aplicaciones que hacen uso de la señal de televisión en directo. Samsung no permite subir estas aplicaciones a su tienda sin un contrato firmado con las cadenas.

En definitiva. Esta primera versión es una versión sólida y usable que, como en todo proyecto, podría obtener mejoras de usabilidad y estabilidad a través de actualizaciones si los responsables y usuarios responden a ella.

Panel de administración

Para poder dar de alta los proyectos es necesario un panel de administración. A continuación se puede ver la web de *analizo.info* y la forma en que se trata esta gestión y visualización de resultados.

La forma en la que se ha integrado el panel de administración en la web de *analizo.info* será tratado de forma más extensa en el proyecto *DESARROLLO DE APLICACIÓN INTERACTIVA PARA TABLET Y SMARTTV EN UN ENTORNO "SECOND SCREEN - (DCADHA-14)*.

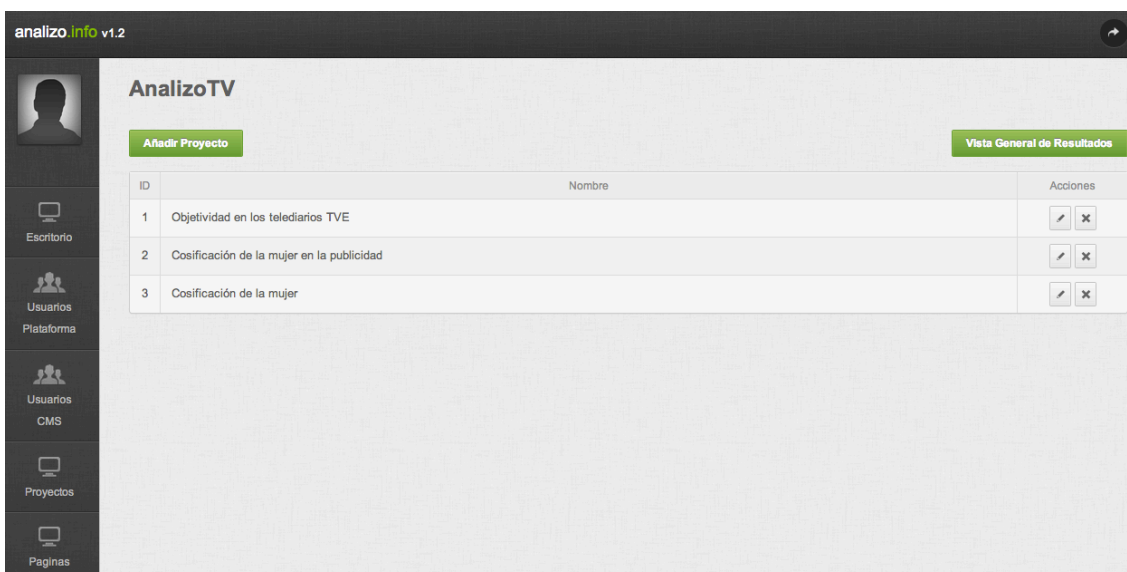


Ilustración 20 – Panel de Administración

Al acceder al panel de administración podemos observar los proyectos que han sido creados en todo momento y podemos editar la información perteneciente a ellos así como borrarlos si es preciso.

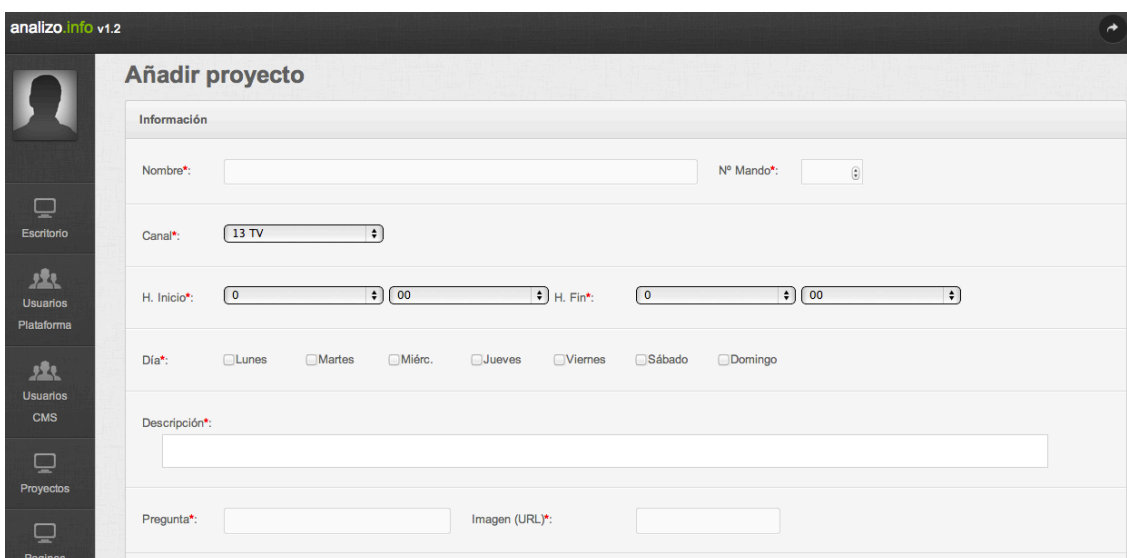


Ilustración 21 – Panel de Administración

Si decidimos crear un nuevo proyecto nos aparecerá un formulario en el que tendremos que rellenar la información pertinente a el proyecto a crear.

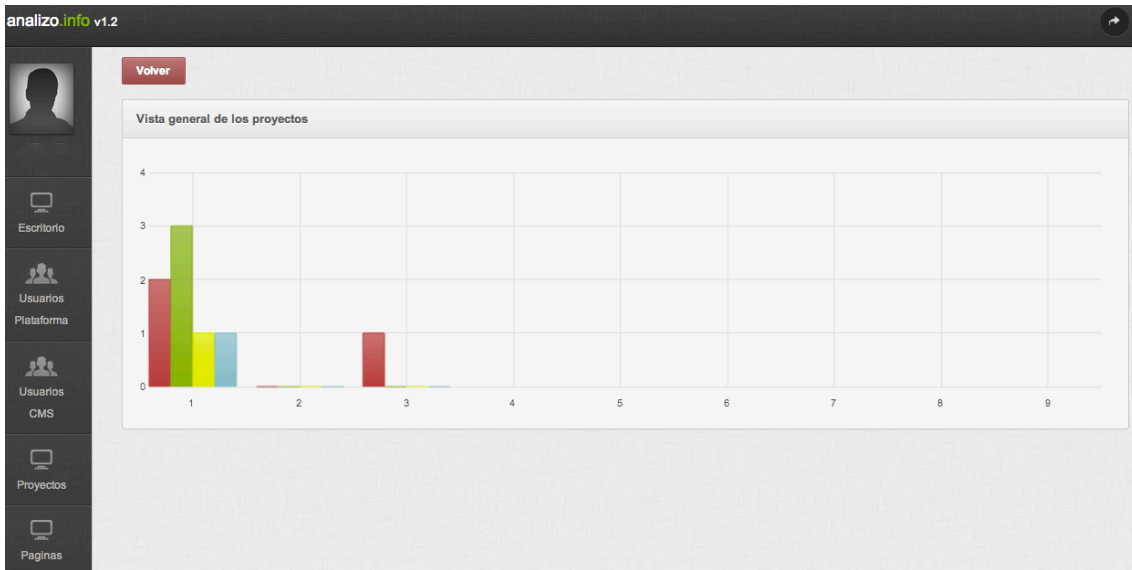


Ilustración 22 - Panel de Administración

Si consultamos la información de los proyectos podemos ver de una manera muy gráfica el estado de los registros procedentes de la aplicación para cada proyecto.

Conclusiones

Al concluir el proyecto propuesto podemos sin lugar a dudas afirmar que ha sido ante todo un reto. El entorno de desarrollo con el que hemos tenido que trabajar es muy nuevo y a pesar de estar familiarizados con ciertas de las herramientas utilizadas, esto ha hecho que nuestra labor a la hora de implementar cambios y funcionalidades haya sido más laboriosa de lo esperado en ciertas etapas. No solo los televisores, sino también la documentación disponible para los desarrolladores, así como en general las posibilidades que ofrece la plataforma están todavía en su etapa inicial. Aún queda contenido que ampliar y añadir al ya disponible, factor acusado a lo largo de nuestra experiencia.

Uno de estos factores afecta de pleno a nuestro proyecto. Una de las tantas políticas de Samsung en relación a la nueva plataforma SmartTV implica restricciones de copyright sobre el uso de la señal de televisión. A efectos prácticos, es posible desarrollar una aplicación para SmartTV que haga uso de la señal emitida por uno o varios canales de televisión. La plataforma en ningún momento restringe esta posibilidad, sin embargo y a fin de evitar violaciones en los derechos de autor, Samsung restringe su publicación en la tienda oficial sin una autorización expresa de los emisores de la señal. De modo que, en nuestro caso y debido a la naturaleza de nuestra aplicación, no podríamos publicarla en la tienda oficial sin conseguir un consentimiento oficial de cada una de las posibles cadenas de cuya señal hacemos uso.

A lo largo del proyecto, siempre hemos optado por una política de código abierto, acorde con la mentalidad de *analizo.info*. Al tratarse de un proyecto real, con una utilidad para una entidad verídica, hemos podido experimentar por primera vez la experiencia de llevar a cabo un proyecto serio de principio a fin. Hemos tenido que tratar con un cliente real, en este caso el equipo de *analizo.info*, también con usuarios reales de la aplicación, además de haber tenido que aprender a llevar y coordinar un desarrollo en equipo.

El hecho de que esta aplicación vaya a tener un uso real nos ha obligado a dar un paso más e integrarla con la web *analizo.info*. Para ello se ha utilizado un panel de control sobre el ya existente de la misma web. Al tratarse de una aplicación de análisis y estadística no tendría mucho sentido no ofrecer una herramienta para estudiar y analizar las mimas. En un principio esta herramienta se creó de forma ajena a la página principal, pero por motivos de comodidad para los usuarios, se integro el panel de control de datos con el ya existe para el resto de proyectos de la web.

Tras 6 meses, no solamente de desarrollo, sino principalmente de investigación y pruebas, nos llevamos con nosotros una experiencia agridulce. Creemos que la plataforma de las SmartTV tiene mucho potencial y desde luego nos hubiera gustado poder experimentar más con ella. Factores como su falta de madurez, carencia de documentación y restricciones han hecho esto imposible. Creemos y esperamos que con el tiempo estos factores disminuyan. Es natural que al aumentar el número de dispositivos también aumenten el número de programadores para ellos y consecuentemente mejore la documentación. Samsung

es conocida desde hace algunos años por sus políticas de código abierto, por lo que sería una lástima que no disminuyeran también las restricciones existentes.

Referencias Bibliográficas

Álvarez, D., Pardo, D. & Rodríguez, M.J. *analizo.info: comunidad online de analistas de información para el cambio social*. Actas III Congreso Universitario sobre redes sociales, Campus de Gandia: Publixd. 21 Feb. 2013. **[Informe]**

EGM. La evolución de la audiencia de Internet en el Estudio General de Medios. Abril–Mayo 2013. Disponible en:
<http://www.aimc.es/-Audiencia-de-Internet-en-el-EGM-.html> **[Informe]**

Beck, K... *Extreme Programming Explained. Embrace Change*. Pearson Education, 1999. Traducido al español como: *Una explicación de la programación extrema. Aceptar el cambio*, Addison Wesley, 2000, pp. 58. Disponible en:
<http://goo.gl/ZURX6m> **[Libro]**

Letelier, P. y Sánchez, E.. *Metodologías Ágiles en el Desarrollo de Software, Taller Metodologías Ágiles en Ingeniería del SW*. DSIC - UPV. JISBD, Alicante. 11 Dec. 2003, pp. 25. Disponible en:
<http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf> **[Material Docente]**

José H. Canós, Patricio Letelier y M^a Carmen Penadés. "Metodologías Ágiles en el Desarrollo de Software." ISG. DSIC - UPV. Valencia. 2 Feb. 2002. Disponible en:
http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf **[Material Docente]**

José Carlos Cortizo Pérez, Diego Expósito Gil y Miguel Ruiz Leyva, *eXtreme Programming*, 2010, pp. 25, 49-50. Disponible en:
<http://www.josek.net/publicaciones/xp.pdf> **[Ensayo]**

Cohn, M.. *User stories applied: for agile software development*. Boston, Mass: Addison-Wesley, 2004. **[Libro]**

Wake, W. C.. *Extreme programming explored*. Boston: Addison Wesley, 2002. **[Libro]**

Newkirk, J. y Martin, R. C.. *Extreme programming in practice*. Boston, MA: Addison-Wesley, 2001. **[Libro]**

Crockford, Douglas. *JavaScript: the good parts*. O'Reilly, 2008. **[Libro]**

Zakas, Nicholas C.. *Professional JavaScript for Web developers*. 3^a edición. Hoboken, N.J.: Wiley, 2012. **[Libro]**

Fowler, M., Foemmel M.. *Continuous Integration*. 2001. Disponible en:
www.martinfowler.com/articles/designDead.html **[Artículo]**

Fowler, M.. *Is Design Dead?*. 2001. Disponible en:
www.martinfowler.com/articles/designDead.html **[Artículo]**

Enlaces de interés

Referencia rápida de comandos de GIT. "*Genbeta Dev*" - Web. 15 Agosto 2013.
<http://www.genbetadev.com/herramientas/referencia-rapida-de-comandos-de-git>

Herramientas que no te pueden faltar como desarrollador web. "*Genbeta Dev*" - Web. 15 Agosto 2013
<http://www.genbetadev.com/desarrollo-web/herramientas-que-no-te-pueden-faltar-como-desarrollador-web>

Seis sencillas buenas prácticas cuando usemos HTML5. "*Genbeta Dev*" - Web. 15 Agosto 2013
<http://www.genbetadev.com/desarrollo-web/seis-sencillas-buenas-practicas-cuando-usemos-html5>

Unas cuantas herramientas que harán maravillas con tus códigos JavaScript. "*Genbeta Dev*" - Web. 15 Agosto 2013
<http://www.genbetadev.com/javascript/unas-cuantas-herramientas-que-haran-maravillas-con-tus-codigos-javascript>

Second-Screen Viewing. "*Mis Apis por tus Cookies*" - Web. 15 Agosto 2013.
www.misapisportuscookies.com/2013/04/second-screen-viewing

Hacia el Second Screen y más allá. "*Mis Apis por tus Cookies*" - Web. 15 Agosto 2013.
<http://www.misapisportuscookies.com/2013/04/hacia-el-second-screen-y-mas-alla/>

Social TV viral. "*Mis Apis por tus Cookies*" - Web. 15 Agosto 2013.
<http://www.misapisportuscookies.com/2013/03/social-tv-viral/>

SDK download, Guide & Forum. "*SAMSUNG SMART TV APPS Developer Forum*" - Web. 15 Agosto 2013
<http://samsungdforum.com>

AllShare keeps all your Samsung smart devices connected "*AllShare Framework*" - Web. 15 Agosto 2013.
<http://developer.samsung.com/allshare-framework>

Transform and innovate with Apps, Data and APIs. "*Apigee*" - Web. 15 Agosto 2013.
<http://apigee.com>

Netflix presenta DIAL, su alternativa universal a AirPlay. "*Gadgets y tecnología: últimas tecnologías en electrónica de consumo - XATAKA*". - Web. 15 Agosto 2013.
<http://www.xataka.com/otros/netflix-presenta-dial-su-alternativa-universal-a-airplay>

La "segunda pantalla", ¿Qué es exactamente". "*Gadgets y tecnología: últimas tecnologías en electrónica de consumo – XATAKA*". - Web. 15 Agosto 2013.
<http://www.xatakahome.com/servicios-de-smart-tv/la-segunda-pantalla-que-es-exactamente>

Easily create apps using the web technologies you know and love: HTML, CSS, and JavaScript." *PhoneGap*". - Web. 15 Agosto 2013.
<http://phonegap.com>

Distraction-free Rich Text Editor for the Web. "*Hallo.js*" - Web. 15 Agosto 2013.
<http://hallojs.org>

Un recurso para desarrolladores de HTML5 para Web abierta. "*HTML5 Rocks*" - Web. 15 Agosto 2013.
<http://www.html5rocks.com/es/>

Pixel Perfect Precision 2 - A PDF Guide On Designing. "*UltraLinx*" - Web. 15 Agosto 2013
<http://theultralinx.com/2013/05/pixel-perfect-precision-2-pdf-guide-designing.html>

Anexo I: Instalación del SDK

Para poder empezar a trabajar es necesario configurar de forma adecuada el entorno de trabajo.

Nosotros nos hemos decantado por el entorno de Windows debido a que el emulador funciona de forma nativa. En los sistemas UNIX/OSX hace uso de una máquina virtual (VirtualBox) para lanzar el emulador, con el gasto de memoria que ello supone.

Requerimientos para instalación

En el desarrollo de este proyecto, se ha optado por la utilización de un entorno virtualizado. De cara al desarrollo el uso de una máquina virtual resulta muy ventajoso, sobre todo de cara a portabilidad y gestión de archivos de datos.

La plataforma de desarrollo necesita de unos mínimos requerimientos para trabajar de forma fluida y cómoda, cabe destacar que estos mínimos son aplicables tanto a máquinas físicas como a entornos virtuales.

La configuración mínima recomendable a nuestro criterio es:

- VMWare fusion 5.0
- Memoria RAM de mínimo 1GB
- Un único procesador de 3GHz o Dual-Core de 1.5GHz
- 20gb de espacio disponible en DD

Cabe destacar que de forma adicional, el uso de las herramientas de desarrollo requiere ciertas actualizaciones, como son:

- .NET Framework 2.0
- Visual C++ Redistributable Package 2005, 2010
- Direct X
- Java 7 update 11

En nuestro caso hemos realizado la instalación en 2 entornos diferentes:

- Ubuntu 12.10 Desktop (32bits)
- Windows 7 Basic (32 bits)

Para un mayor detalle de los requisitos SW para la instalación de las herramientas de desarrollo se puede revisar la documentación disponible en:

<http://developer.samsung.com/develop>

Preparación para la instalación

Antes de realizar la instalación se debe realizar una serie de pasos previos:

- Asegurar que se cumplan los requerimientos mínimos de sistema
- Asegurar que se cuentan con permisos mínimos adecuados para la correcta instalación de todos los paquetes

Instrucciones de instalación

Entorno Linux

Una lista parcial de las versiones soportadas:

- Ubuntu 8.04 LTS, 10.04, 11.04, 11.10, 12.04, 12.10
 - Debian 6.0, 7.0
 - openSUSE 11.4, 12.1, 12.2
 -
1. Lo primero es descargar el SDK y emulador de Samsung para Virtual Box desde: <http://www.samsungdforum.com/devtools/sdkdownload> (Nos pedirá que nos registremos antes de proceder con la descarga)
 2. Instalar última versión disponible de Java:
<http://www.java.com/es/download/manual.jsp>

```
//Añadimos el repositorio de Java
sudo add-apt-repository
ppa:webupd8team/java
//Actualizamos nuestra lista de
repositorios
sudo apt-get update
//Instalamos Java
sudo apt-get install oracle-java7-
installer
```

3. Instalar Virtual Box, ya que el emulador de SamsungTV para sistemas Unix hace uso de esta aplicación:
https://www.virtualbox.org/wiki/Linux_Downloads

```
sudo apt-get install dkms
sudo apt-get install virtualbox-4.2
sudo apt-get -f install -> las
dependencias
```

4. Instalar el emulador de SamsungTV para Linux. El emulador de SmartTV debe ser importado a Virtual Box una vez instalado, a continuación se detalla cómo:
 - Iniciar el programa Virtual Box
 - Seleccionar Archivo -> Importar aparato

- En el menú de importación, seleccionar el archivo del emulador (Samsung_Smart_TV_Emulator.ova) que nos hemos descargado en los pasos previos:

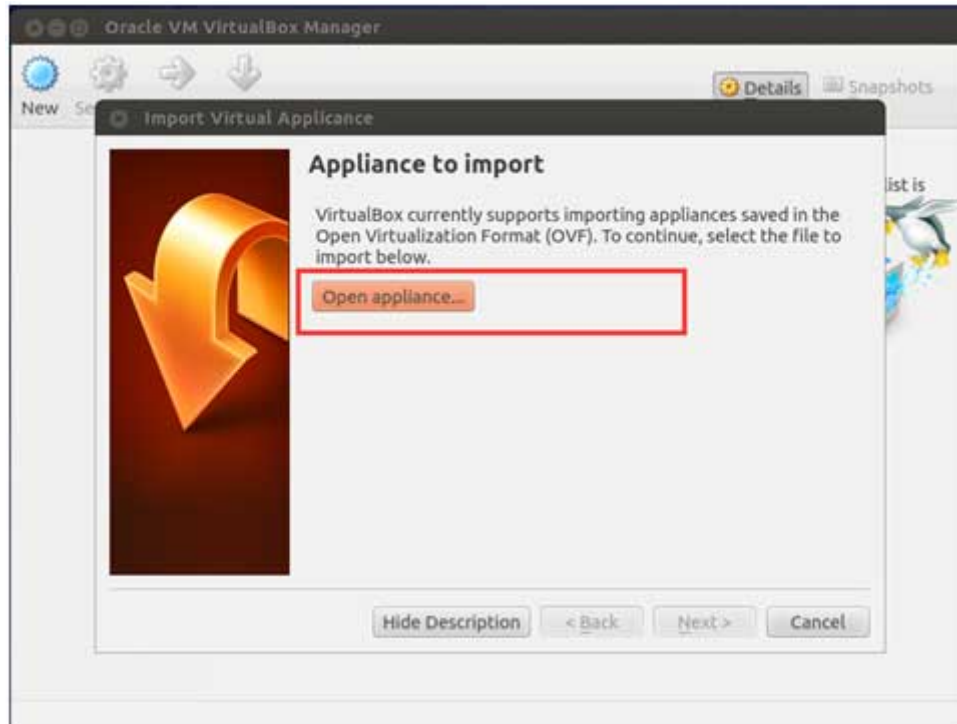


Ilustración 23 - Anexo I. Instalación Unix

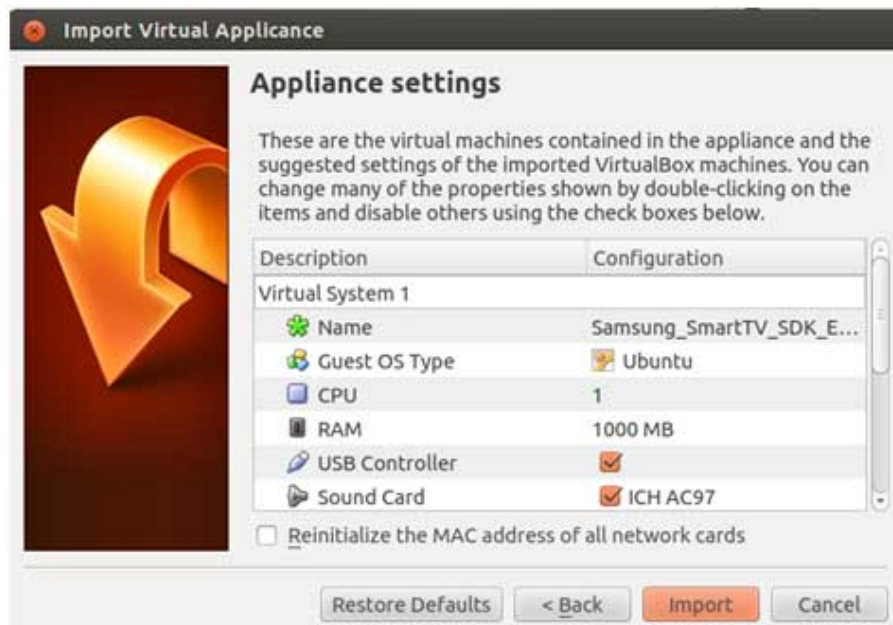


Ilustración 24 - Anexo I. Instalación Unix

- La máquina virtual del emulador se cargará:




Ilustración 25 - Anexo I. Instalación Unix

- Una carpeta compartida es usada para acceder a las aplicaciones del IDE y el entorno de emulación. Para configurar dicha carpeta deberemos hacer clic derecho sobre el emulador y seleccionar “Propiedades”:



Ilustración 26 - Anexo I. Instalación Unix

- En la pestaña de carpetas compartidas, haremos clic sobre  y añadiremos la carpeta de APPs que se incluía en la carpeta de instalación del Samsung Smart TV SDK. Seleccionar “Automontar” y OK:

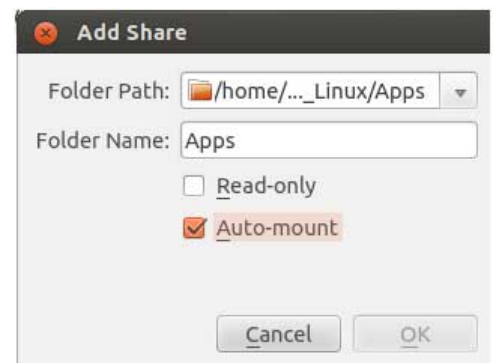
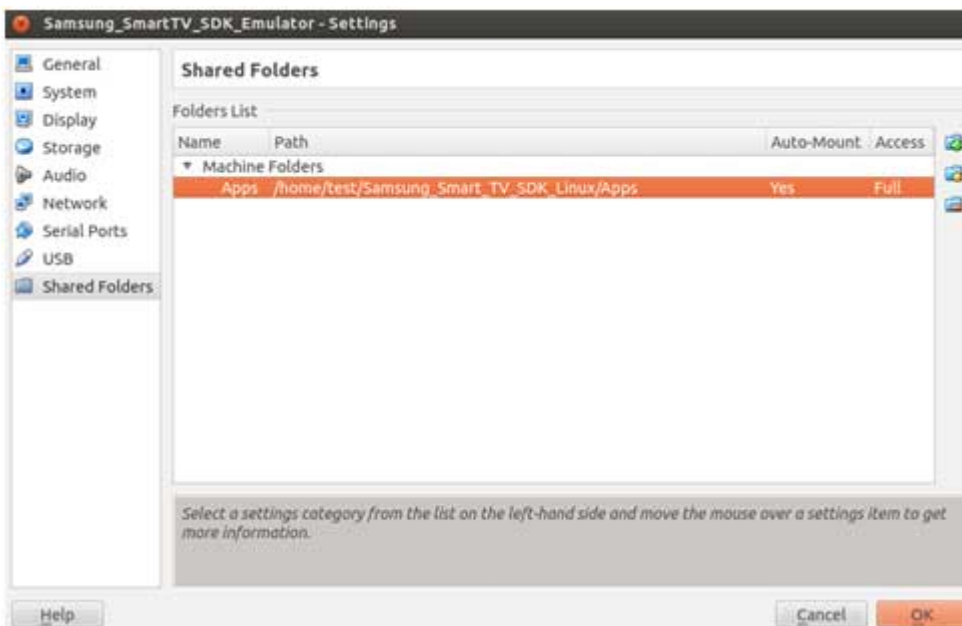


Ilustración 27 - Anexo I. Instalación Unix

- El entorno de Virtual Box ya estará configurado, tan solo queda iniciarlo :

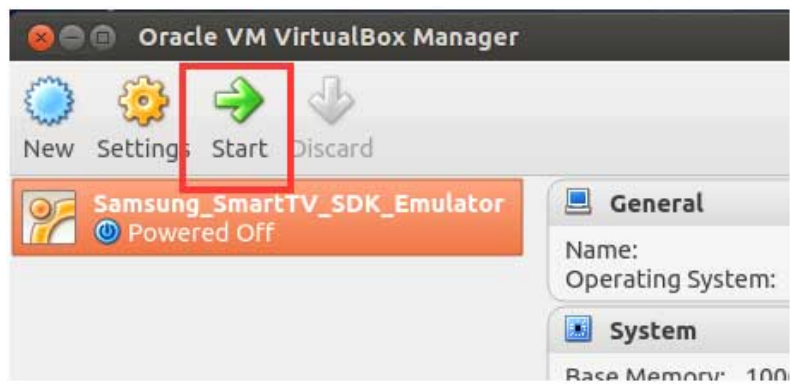


Ilustración 28 - Anexo I. Instalación Unix

- Tras esto, solo nos queda abrir Eclipse - Oracle y crear nuestro directorio de trabajo

Instalación Virtual Box:

<https://www.virtualbox.org/manual/ch02.html>

Instalación SDK Linux:

<http://www.samsungdforum.com/Guide/d09/index.html>

Guía uso emulador Smart TV:

<http://www.samsungdforum.com/Guide/d07/index.html>

Entorno Windows

Versiones soportadas (32 bits):

- Windows XP – Service Pack 2
- Windows 7

1. Lo primero es descargar el SDK y emulador de Samsung para Virtual Box desde:
<http://www.samsungdforum.com/devtools/sdkdownload>
(Nos pedirá que nos registremos antes de proceder con la descarga)
2. Instalar última versión disponible de Java:
<http://www.java.com/es/download/manual.jsp>
3. Instalamos complemento Direct X:
<http://www.microsoft.com/download/en/details.aspx?id=35>
4. Instalar el emulador de SamsungTV para Windows:
 - Para ello, empezamos ejecutando el archivo descargado en el primer paso. Nos soltará que seleccionemos idioma y procederá a darnos la bienvenida y mostrarnos el acuerdo de licencia.
 - Al seleccionar los componentes a instalar debemos seleccionar tanto el editor como el emulador. A continuación nos pedirá una carpeta de destino:



Ilustración 29 - Anexo I. Instalación Windows

- El SDK hace uso del servidor HTTP Apache, si no está instalado con anterioridad simplemente debemos acceder a instalarlo durante la instalación:

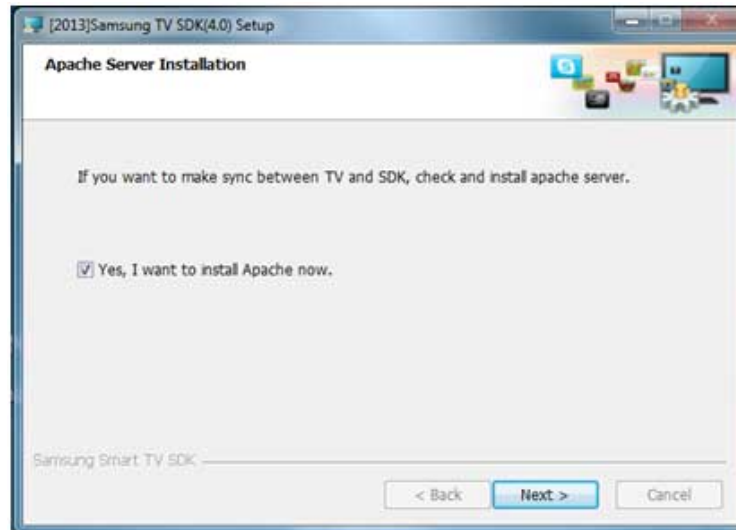


Ilustración 30 - Anexo I. Instalación Windows

- Este proceso instalará Visual C++ Redistributable Package 2005-2010 si no lo tenemos instalado, el IDE Eclipse y los emuladores de los distintos modelos de televisión Samsung. Al finalizar deberíamos tener disponibles los siguientes iconos:

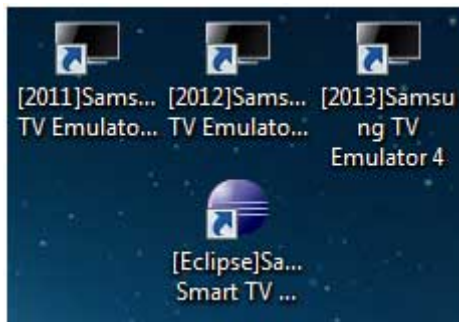


Ilustración 31 - Anexo I. Instalación Windows

5. Una vez concluida la instalación del SDK de Samsung para *SmartTV* procederemos a descargar el SDK genérico para Android:
<http://developer.android.com/sdk/index.html>

6. Tras esto, abriremos Eclipse - Oracle y procederemos a instalar el SDK genérico de Android junto con el emulador de la Samsung Galaxy Tab. Seguiremos la ruta: Help ->Install, donde seleccionaremos la opción “Add” e instalaremos el repositorio correspondiente, disponibles en:

<http://dl.google.com/eclipse/android/eclipse>

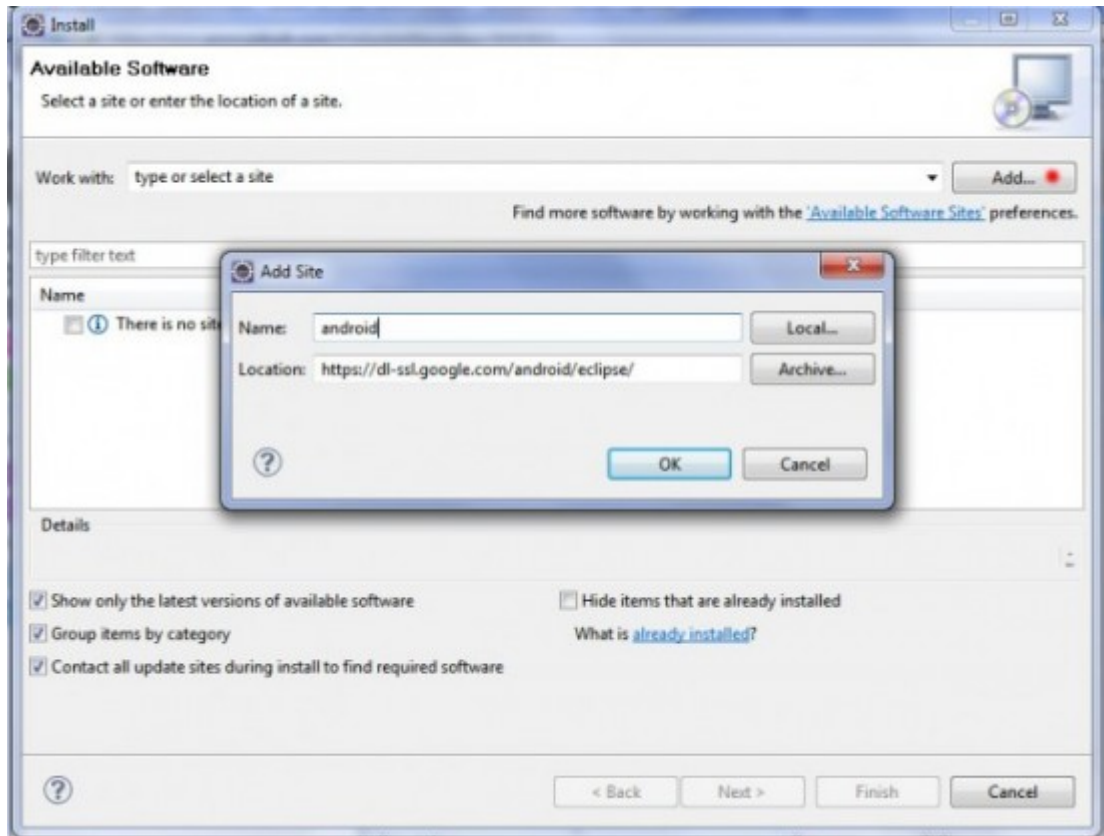


Ilustración 32 - Anexo I. Instalación Windows

7. Seguidamente se nos solicitará que reiniciemos el programa. Una vez reiniciado, debemos configurar Eclipse para que acceda al SDK de Android ya instalado. Para ello, abrimos el menú “Window” y seleccionamos “Preferences”.

8. En el apartado *Android => SDK Location* debemos introducir la ruta donde hemos descomprimido el SDK de Android (genérico) descargado con anterioridad:

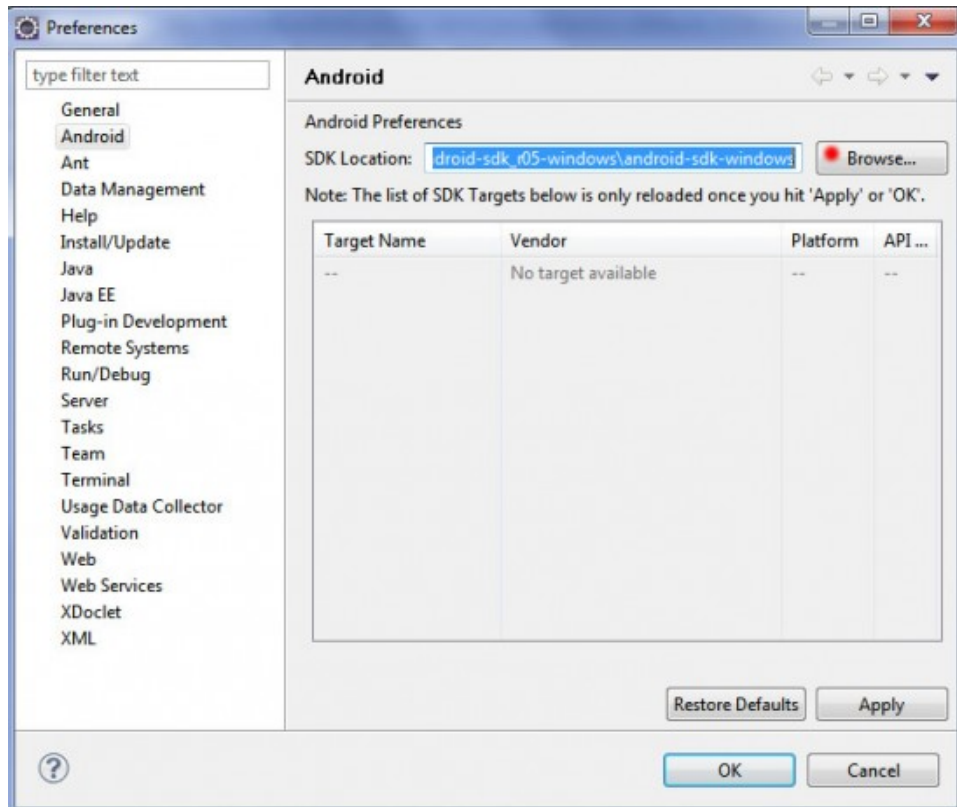


Ilustración 33 - Anexo I. Instalación Windows

Nota: En *Windows => Android SDK and AVD Manager* podemos descargar actualizaciones, documentación y ejemplos.

9. Tras esto, solo nos queda abrir Eclipse - Oracle y crear nuestro directorio de trabajo

Instalación SDK Windows:

<http://www.samsungdforum.com/Guide/d11/index.html>

Guía uso emulador Smart TV:

<http://www.samsungdforum.com/Guide/d07/index.html>

Creación de nuestro directorio de trabajo

Una vez seguidos los anteriores pasos de instalación, podemos proceder a empezar a trabajar con Eclipse. Para ello seguiremos los siguientes pasos:

1. Ir a la ruta Windows -> Preferences
2. Expandir Run/Debug -> Launching -> Launch Configurations
3. Seleccionar el check box "Filtered checked launch configuration types"
4. Ir al final de la lista y elegir "XSL"
5. Seleccionar aplicar

Nota: Para que el entorno funcione correctamente puede ser necesario eliminar el archivo *.out.xml ANTES de seleccionar el botón Run (arrancar entorno).


Anexo II: Instalación de la aplicación en la TV

En el momento de instalar el SDK en las plataformas Windows, este nos da la opción de instalar un servidor Apache que nos facilita el testeo de nuestras aplicaciones en la Samsung SmartTV.

Para ello necesitamos hacer uso del usuario especial “develop” que nos pedirá la dirección IP de la máquina que usamos para el desarrollo, en este caso se trata de una máquina Windows por las razones que indicamos en el **Anexo I**.

Preparando el servidor

La forma sencilla de preparar el lado del servidor es a través de eclipse.

Cuando empaquetamos una aplicación a través del botón , tenemos la opción de dejarla preparada en el servidor como podemos ver en la Ilustración 1 de este anexo.

De esta forma el sistema nos creará en la raíz de la unidad C: el fichero widgetlist.xml junto a la carpeta Widget que contendrá nuestra aplicación.

Podemos reproducir esta estructura en cualquier servidor, no tiene porqué ser la máquina con la que trabajamos. Los únicos requisitos son que el servidor este accesible desde el puerto 80 y el fichero widgetlist.xml se encuentre en la raíz del mismo.

La estructura del fichero widgetlist.xml y puede ser editada a nuestra conveniencia:

```
<?xmlversion="1.0" encoding="UTF-8" standalone="no"?>
<rspstat="ok">
<list>
<widget id="SmartTweets">
<title>SmartTweets</title>
<compressionsize="21312" type="zip"/>
<description/>
<download>http://192.168.1.2/Widget/SmartTweets_0.400_America_20130508.zip</download>
</widget>
</list>
</rsp>
```

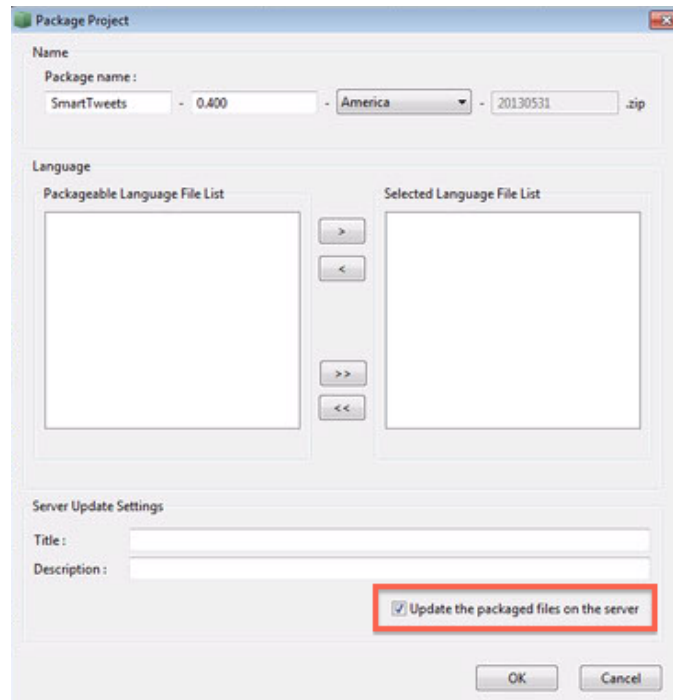



Ilustración 34 - Anexo II. Configuración Servidor

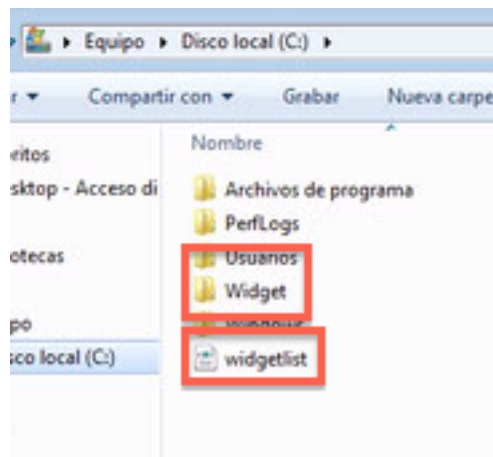


Ilustración 35 - Anexo II. Configuración Servidor

Preparando la SmartTV

Como hemos dicho anteriormente, lo primero es activar el usuario “develop” que todas las televisiones de *Samsung* traen por defecto.

Para ello lo primero que haremos será salir de nuestra cuenta personal e ingresar con una nueva cuenta a través de la opción *Login*.



Ilustración 36 - Anexo II. Configuración SmartTV

Cuando aparezca el panel de inicio de sesión debemos poner “**develop**” en el ID y veremos que el campo de contraseña se rellena solo. Tan solo debemos pulsar en *Login*.



Ilustración 37 - Anexo II. Configuración SmartTV

Si todo ha ido bien veremos la palabra **develop** en la esquina inferior izquierda de la pantalla.

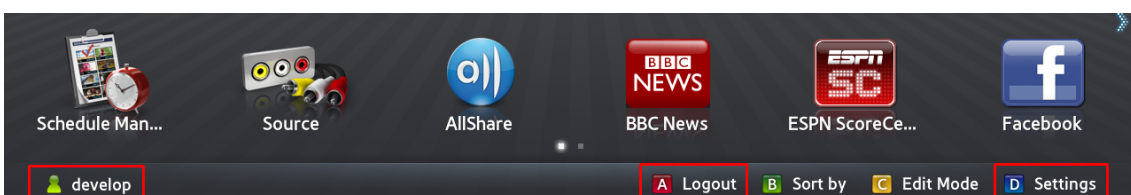


Ilustración 38 - Anexo II. Configuración SmartTV

Para sincronizar nuestras aplicaciones solo nos queda seguir la ruta *herramientas* -> *config.* -> *desarrollo*, aceptar la licencia e introducir la IP de la máquina que hemos configurado en el apartado anterior. Hay que recordar que la máquina ha de tener disponible el puerto 80 puesto que el menú de Samsung no nos da la opción de especificar puerto.

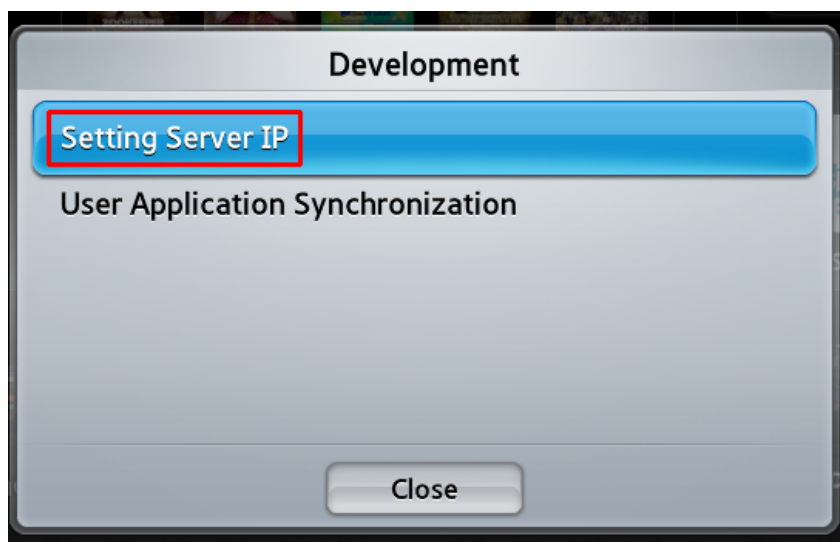


Ilustración 39 - Anexo II. Configuración SmartTV

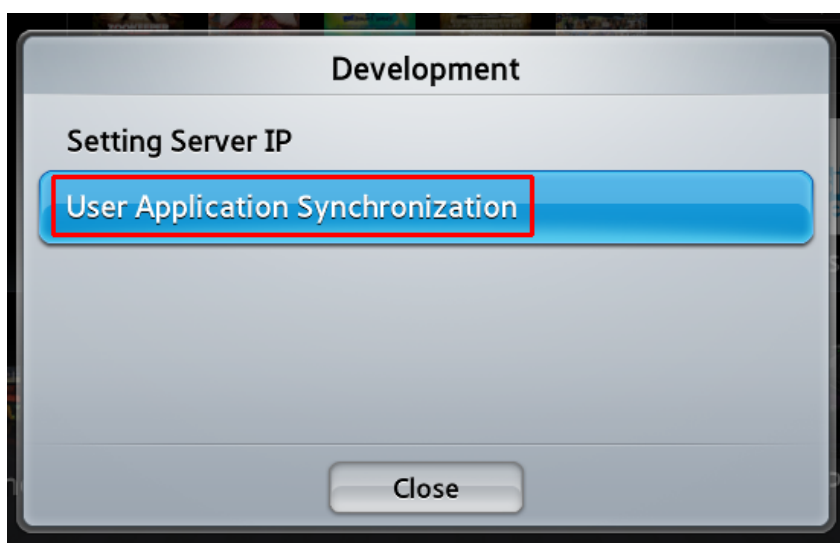


Ilustración 40 - Anexo II. Configuración SmartTV

Solo nos queda sincronizar las aplicaciones y si todo está bien y el servidor está disponible, nuestra aplicación aparecerá como una de las disponibles en el menú principal del *SmartHub* y podremos proceder a su testeo.

Anexo III: Apigee vs. API propia

Apigee es un servicio de *Transferencia de Estado Representacional*, también conocido como API REST. Estos servicios se caracterizan por tener un conjunto de operaciones bien definidas que se aplican a todos los recursos de información. Mediante las peticiones POST, GET, PUT y DELETE de HTTP conseguimos acceder, actualizar o eliminar los datos con los que trabajamos de forma muy sencilla y desde cualquier dispositivo, sin necesidad que este tenga que abrir una conexión *MySQL* ni nada por el estilo. Simplemente intercambiando un objeto XML o JSON como en este caso, conseguimos la persistencia de nuestra información.

En este proyecto nos hemos decantado por hacer uso del servicio *Apigee* para gestionar la información que manejamos referente a proyectos y usuarios. La principal razón detrás de esta elección es que *Apigee* nos ofrece un almacenamiento y gestión de datos y usuarios ilimitado de forma gratuita, esto nos permite demostrar el potencial de nuestro proyecto sin que suponga ningún gasto para el cliente, *analizo.info* en este caso.

Además cuenta con librerías en múltiples lenguajes que permiten una fácil interacción con el servicio y nos facilita el desarrollo del proyecto.

Es un servicio muy potente que tiene características muy interesantes, como pueden ser:

- Gestión de usuarios.
- Restricciones de vistas.
- Notificaciones Push.
- Características sociales como feeds de actividades, relaciones entre usuarios...
- Modelado de datos.
- Geolocalización.
- Personalización de eventos.
- Métricas y captura de errores.
- Gestión de distintas aplicaciones.
- Transformación de XML a JSON y viceversa.

Como puede observarse, nos proporciona todas las utilidades básicas y muchas más de las que hemos hecho uso.

Pese a ser un servicio muy potente a tener en cuenta es cierto que, por lo menos en su versión gratuita, *Apigee* no nos permite manipular nuestra API al nivel de detalle que nos gustaría. Por ejemplo, si queremos que al registrarse un usuario se envíe un mail de confirmación o gestionar una recuperación de contraseña con *Apigee* de momento no es posible.

Por esta razón proponemos a *analizo.info* el desarrollo de una API propia.

El desarrollo de una API supone un coste de tiempo que no podíamos afrontar durante este proyecto. Esto no quita que en las próximas líneas expongamos algunos consejos que pueden ser de utilidad si se decide llevar a cabo el proyecto.

Node.js

Node.js es un entorno JavaScript de lado de servidor, por lo que no tenemos que aprender un nuevo lenguaje de programación, que utiliza un modelo asíncrono y dirigido por eventos.

Node.js usa el motor de JavaScript V8 de Google: una Máquina Virtual (VM) tremendamente rápida y de gran calidad escrita por gente como Lars Bak, uno de los mejores ingenieros del mundo especializados en VMs. No olvidemos que V8 es actualizado constantemente y es uno de los intérpretes más rápidos que puedan existir en la actualidad para cualquier lenguaje dinámico. Además las capacidades de Node.js para I/O (Entrada/Salida) son realmente ligeras y potentes, dando al desarrollador la posibilidad de utilizar la I/O del sistema con una gran carga.

Soporta protocolos TCP, DNS y http y a diferencia de Apache no crea un nuevo hilo por cada conexión cliente-servidor.

Además Node.js es completamente modular y cuenta con un gran apoyo de la comunidad. Hay cientos de módulos ya creados para gestionar usuarios, conectar con bases de datos, hacer uso de fuentes sociales de forma sencilla...

Para ver cómo funciona Node.js se puede consultar el blog de Carlos Azaustre dónde podemos ver un ejemplo realmente útil para nuestro propósito. En el nos explica cómo podemos obtener una API REST completa en apenas unos minutos:

<http://carlosazaustre.es/blog/como-crear-una-api-rest-usando-node-js/>

Quizá parezca que meterse en el desarrollo de algo tan “nuevo” no merezca la pena, pero ya son mucho los servicios basados en node.js y estamos seguro de que cada vez lo serán más. La web de *LinkedIn*, el *New York Times* o *eBay*³ ya utilizan esta tecnología.

³ <http://nodejs.org/industry>