



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

TEMA: Ampliación del alcance de los servicios web de la Universidad Tecnológica Israel por medio de dispositivos móviles.

**AUTORES: Diego Edmundo Román Noboa
Marcelo Xavier Secaira Larco**

TUTOR / A: Mg. WILMER RAMIRO VALLE BASTIDAS

AÑO: 2017

Datos generales:

Tema:	Ampliación del alcance de los servicios web de la Universidad Tecnológica Israel por medio de dispositivos móviles
Estudiantes:	Diego Edmundo Román Noboa Marcelo Xavier Secaira Larco
Carrera:	Ingeniería de sistemas informáticos
Tutor:	Mg. Wilmer Ramiro Valle Bastidas
Fecha:	2017-07-31

DEDICATORIA

A mi familia, amigos y compañeros, con especial sentimiento de gratitud a mis queridos abuelos José y María cuyas palabras de estímulo y empuje han sido el soporte para continuar con mis estudios.

A mi hermana Gabriela y a todos quienes de una u otra forma me han apoyado durante todo el proceso de estudio.

Siempre apreciaré todo lo que han hecho.

Diego Román.

A Dios por las incalculables bendiciones sobre mí, que me han permitido llegar a cumplir una meta más.

A mis pilares fundamentales quienes me impulsan a seguir día tras día, mi madre por su constancia, apoyo y sacrificio fuente de luz y amor, mi hermana y sobrina Dulce Romina; A mi hijo Alejandro Sebastián quien me hizo mirar la vida de una forma distinta, a quien entrego mi corazón.

A REDIME CIA LTDA. Por su apoyo constante en todo mi proceso universitario.

A todos quienes formaron parte de este gran reto. Gracias infinitas.

Marcelo Secaira L.

AGRADECIMIENTOS

A Ing. Wilmer Valle, por su guía y colaboración en todo el proceso de desarrollo de la presente tesis.

A compañeros y maestros quienes han aportado de distintas maneras, para el desarrollo de la primera aplicación móvil de la Universidad Tecnológica Israel.

RESUMEN

En la actualidad, resulta inconcebible un día a día sin teléfono móvil, y más en concreto, sin Smartphone. Debido al auge de estos dispositivos, cada vez más personas utilizan los múltiples servicios que ofrecen. IOS y Android son los sistemas operativos para dispositivos móviles que ha experimentado mayor crecimiento en los últimos años. Por otro lado, la integración de los sistemas por medio de los Web Services, es otro de los aspectos más presentes hoy en día. Por ello aunar estas fuerzas – Web Services y Smartphone- en las instituciones cobra cada vez más sentido. En esta dirección, este proyecto puede realizar una aportación a la Universidad Tecnológica Israel.

El presente proyecto está diseñado para aumentar el alcance de los servicios web por medio de aplicaciones para teléfonos inteligentes. Por lo que una aplicación móvil se hace esencial con el objetivo de otorgar comodidad a los usuarios en este caso los alumnos de la Universidad. Este proyecto intenta ofrecer una aplicación que permite tanto la consulta de servicios, como la publicación de noticias de interés para la comunidad universitaria.

A lo largo del documento se presenta la aplicación realizada, exponiendo una descripción de características y requisitos principales, todas las tecnologías utilizadas en el desarrollo de la herramienta, tanto del lado cliente como del lado del servidor, así como un análisis que recorrerá todas las fases del proyecto desde el diseño hasta la implementación.

ABSTRACT

At present, it is inconceivable a day to day without mobile phone, and more specifically, without Smartphone. Due to the rise of these devices, more and more people use the many services they offer. IOS and Android are the operating systems for mobile devices that has experienced the greatest growth in recent years. On the other hand, the integration of the systems through the Web Services, is another of the most present aspects today. For this reason, joining these forces - Web Services and Smartphone - in the institutions becomes more and more sense. In this direction, this project can make a contribution to the Israel Technological University.

This project is designed to increase the reach of web services through smartphone applications. A mobile application becomes essential with the aim of providing comfort to users in this case students of the University. This project tries to offer an application that allows both the consultation of services, and the publication of news of interest for the university community.

Throughout the document the application is presented, presenting a description of characteristics and main requirements, all the technologies used in the development of the tool, both client and server side, as well as an analysis that will cover all phases from design to implementation

ÍNDICE

SECCIÓN I.....	1
PROBLEMA DE INVESTIGACIÓN	1
OBJETIVO GENERAL.....	1
OBJETIVOS ESPECÍFICOS	1
INTRODUCCIÓN	1
HIPÓTESIS.....	2
SECCIÓN II	3
MARCO TEÓRICO Y CONCEPTUAL	3
INTRODUCCIÓN	3
ANDROID	4
APLICACIONES DE MAQUINA VIRTUAL DALVILK Y APLICACIONES ANDROID	4
ESTRUCTURA ANDROID, JAVA Y XML.....	5
VERSIONES DE ANDROID.....	5
ENTORNO DE DESARROLLO.....	5
ANDROID SDK.....	6
ANDROID STUDIO	6
DISPOSITIVO VIRTUAL DE ANDROID	7
VISTA DE PROYECTOS DE ANDROID	7
SERVICIO WEB	9
IOS	12
XCODE.....	14
SWIFT.....	14
JAVA.....	15
XML.....	15
JSON	16
PHP	16
PRINCIPALES USOS	16
MYSQL	17
SQLITE.....	17

SECCIÓN III.....	18
METODOLOGÍA.....	18
METODOLOGÍA DE INVESTIGACIÓN.....	18
OBJETO DE ESTUDIO.....	19
ANÁLISIS E INTERPRETACIÓN DE LA ENCUESTA.....	19
DIAGNOSTICO.....	25
METODOLOGÍA DE DESARROLLO.....	25
PROPUESTA	27
Exploración.....	27
Inicialización	31
Producción.....	37
Estabilización.....	78
Pruebas y arreglos del sistema.....	83
SECCIÓN IV.....	86
CONCLUSIONES.....	86
RECOMENDACIONES	87
BIBLIOGRAFÍA.....	88

Índice de Gráficos

Figura 1 Resultado encuesta 1	20
Figura 2 Resultado encuesta 2	21
Figura 3 Resultado encuesta 3	21
Figura 4 Resultado encuesta 4	22
Figura 5 Resultado encuesta 5	22
Figura 6 Resultado encuesta 6	23
Figura 7 Resultado encuesta 7	23
Figura 8 Resultado encuesta 8	24
Figura 9 Resultado encuesta 9	24
Figura 10 Arquitectura de Servicios Web tipo REST.....	29
Figura 11 Inicialización del proyecto	31
Figura 12 Preparación del proyecto	31
Figura 13 Planificación del día inicial	33
Figura 14 Arquitectura Base	34
Figura 15 Modelo Relacional.....	35
Figura 16 Día de trabajo inicial	36
Figura 17 Etapas de producción.....	37
Figura 18 Día de planificación.....	37
Figura 19 Día de trabajo	38
Figura 20 Día de liberación.....	39
Figura 21 Ingreso a consola	41
Figura 22 Menú principal.....	41
Figura 23 Envío de notificaciones	42
Figura 24 Configuración noticias.....	42
Figura 25 Configuración servicios.....	42
Figura 26 Configuración Galería	43
Figura 27 Visualización de noticias	53
Figura 28 Visualización de servicios	53
Figura 29 Visualización de galería	54
Figura 30 Visualización de redes sociales	54

Figura 31 Visualización de información.....	55
Figura 32 Vista inicial IOS	66
Figura 33 Vista de Noticias IOS	66
Figura 34 Visualización de servicios IOS.....	67
Figura 35 Visualización Galería IOS.....	67
Figura 36 Redes Sociales IOS.....	68
Figura 37 Información IOS.....	68
Figura 38 Día de planificación.....	78
Figura 39 Día de trabajo	80
Figura 40 Documentación.....	81
Figura 41 Día de liberación.....	82
Figura 42 Pruebas y arreglos del sistema.....	83

Índice de Tablas

Tabla 1 Partes Interesadas.....	27
Tabla 2 Asignación del personal.....	28
Tabla 3 Lista de verificación.....	29
Tabla 4 Recursos Físicos	31
Tabla 5 Recursos Técnicos	32
Tabla 6 Pruebas de aceptación.....	36
Tabla 7 Ingreso a consola de administración	43
Tabla 8 Configuración de noticias	44
Tabla 9 Configuración de servicios	45
Tabla 10 Configuración de galería.....	45
Tabla 11 Publicación del sitio web	46
Tabla 12 Pruebas ingreso a la consola de administración.....	47
Tabla 13 Pruebas ingresos de noticias	48
Tabla 14 Pruebas configuración de servicios.....	48
Tabla 15 Pruebas configuración de galería.....	49
Tabla 16 Verificación de auditorías.....	51
Tabla 17 Listado de deficiencias.....	52
Tabla 18 Vista de inicio	55
Tabla 19 Vista de noticias.....	56
Tabla 20 Vista de servicios	56
Tabla 21 Vista de galería	57
Tabla 22 Vista redes sociales	58
Tabla 23 Vista de información.....	59
Tabla 24 Pruebas vista de noticias	59
Tabla 25 Pruebas vista de servicios	60
Tabla 26 Pruebas vista de galería.....	61
Tabla 27 Pruebas vista redes sociales	61
Tabla 28 Pruebas vista de información.....	62
Tabla 29 Verificación de auditorías.....	64
Tabla 30 lista de resumen de deficiencias.....	65

Tabla 31 Vista de inicio	69
Tabla 32 Vista de noticias.....	69
Tabla 33 Vista de servicios.....	70
Tabla 34 Vista de galería	71
Tabla 35 Vista redes sociales.....	71
Tabla 36 Vista de información.....	72
Tabla 37 Pruebas de aceptación.....	73
Tabla 38 Verificación de auditorias.....	77

SECCIÓN I

PROBLEMA DE INVESTIGACIÓN

La Universidad Tecnológica Israel actualmente pone a disposición la consulta de sus servicios estudiantiles como: consulta de notas, carga académica, etc. a través de aplicativos webs, por lo cual la falta de una aplicación móvil limita a los alumnos al uso de un ordenador para consultar la información que requieran.

Además en la actualidad la mayoría de universidades en el Ecuador disponen de aplicaciones móviles para la consulta de sus servicios estudiantiles, brindando a sus alumnos facilidad y disponibilidad para acceder a los mismos. Por estas razones la Universidad Tecnológica Israel no puede estar sin tomar ventaja de esta tecnología y debe poseer su aplicación móvil.

OBJETIVO GENERAL

Desarrollar una aplicación para los usuarios de teléfonos inteligentes iPhone y Android los cuales podrán acceder a los servicios web de la Universidad Tecnológica Israel

OBJETIVOS ESPECÍFICOS

- Desarrollar aplicaciones móviles nativas para las plataformas IOS y Android.
- Implementar el desarrollo de la consola de administración web para el manejo de la información que será visualizada en las aplicaciones móviles.
- Ampliar el conocimiento de servicios web para desarrollar una interfaz REST de comunicación e intercambio de datos entre las plataformas.

INTRODUCCIÓN

El avance de tecnología y de las telecomunicaciones ha creado nuevas vías de comunicación social, en nuestro medio es evidente el crecimiento del uso de teléfonos inteligentes

facilitando el acceso a cualquier tipo de información disponible en el internet, produciendo sociedades inteligentes.

En la actualidad las aplicaciones móviles son la puerta de acceso de información de las instituciones y empresas que desean ofrecer sus servicios por medio de la web. El desarrollo de aplicaciones móviles se ha convertido en una poderosa herramienta que puede aprovechar los servicios que posee un teléfono inteligente, por ejemplo, el uso de servicios de geolocalización por medio del GPS, también se puede capturar y almacenar fotografías y videos por medio de la cámara que poseen.

Uno de los servicios que causó un salto cualitativo en cuanto a teléfonos inteligentes es el poder tener acceso a internet en cualquier lugar donde la cobertura de su operadora telefónica les permita. En los últimos años han aparecido varios fabricantes de teléfonos los cuales esta divididos por el sistema operativo que les permite funcionar. Los principales son Windows móvil, BlackBerry OS, IOS y Android. Los más utilizados actualmente son IOS del fabricante APPLE y Android de GOOGLE el que puede funcionar en varios dispositivos de diferentes fabricantes.

El presente trabajo de investigación es para mejorar la comunicación de información de interés para los alumnos de la Universidad Tecnológica Israel por medio del uso de teléfonos inteligentes de diferentes marcas y las más usadas en la actualidad.

HIPÓTESIS

“La aplicación móvil de la Universidad Tecnológica Israel mejorará la cobertura y facilitará el acceso a los servicios de su portal web”

SECCIÓN II

MARCO TEÓRICO Y CONCEPTUAL

INTRODUCCIÓN

El ordenador ha estado en constante evolución desde mediados del siglo XX. Continuó consiguiendo ser más pequeño en tamaño, usando menos energía y realizando cálculos más avanzados. En el 2007 Apple lanzó su iPhone para lograr el próximo objetivo en la informática. Este nuevo tipo de herramienta de comunicación, llamada Smartphone, tal como se denomina generalmente a un teléfono inteligente que además es una computadora de mano. Dicho dispositivo fue utilizado por primera vez en 1992, Apple fue la primera empresa en lanzar un Smartphone a un público más amplio. Esta evolución está liderada por fabricantes de computadoras y empresas de software (Hall & Anderson, 2009).

Un competidor de Apple iPhone OS es el sistema operativo Android; dicho sistema se origina en un pequeño software de una empresa adquirida por Google y ahora es propiedad de Open Handset Alliance (OHA), donde Google es un miembro. OHA cuenta con más de un centenar de empresas miembros como: operadores móviles, empresas de semiconductores, fabricantes de teléfonos, empresas de software y empresas de comercialización. Impulsado a través de la licencia de Apache, cualquier persona puede usar el kit de desarrollo de software de Android (SDK) para desarrollar aplicaciones que pueden ejecutarse en el sistema operativo en mención. Un punto interesante del sistema operativo es el uso de técnicas comunes no propietarias, como el lenguaje de programación Java en combinación con Extensible Markup Language (XML). Esto lo hace abierto, simple y fácil de usar para una parte de la comunidad de desarrolladores

La tesis refleja experiencias de uso de XML, el lenguaje de programación Java y SDK para Android.

Esta tesis se centra en el desarrollo de las aplicaciones nativas para ANDROID e IOS que serán ejecutadas en los dispositivos móviles de estas dos plataformas, además la implementación de ambientes de desarrollo que dispongan de todas las herramientas apropiadas para programar y

compilar el código en ambas tecnologías. Finalmente describiremos todos los métodos y procesos necesarios que servirán específicamente para desarrollar e implementar la primera aplicación oficial de la Universidad Tecnológica Israel.

ANDROID

Android es un sistema de código abierto que puede ser utilizado por cualquier persona. Un fabricante de teléfonos móviles puede usar Android si siguen el acuerdo indicado en el Software Development Kit, el fabricante de dispositivos móviles puede ocupar todas las utilidades y herramientas para interconectar el hardware del dispositivo y el software, la única restricción que los fabricantes tienen es no realizar ninguna modificación al Kernel de Linux.

Android es un entorno de software y no una plataforma de hardware, que incluye un sistema operativo, Linux kernel based OS la cual aloja la máquina virtual Dalvik. La máquina virtual Dalvik funciona con aplicaciones de Android como instancias de la máquina virtual. Android contiene una interfaz de usuario muy robusta además de las bibliotecas de clases Java y soporte multimedia.

APLICACIONES DE MAQUINA VIRTUAL DALVILK Y APLICACIONES ANDROID

Cada aplicación de Android se ejecuta en su propia máquina virtual Dalvik y las aplicaciones desarrolladas no están subordinadas a las aplicaciones incorporadas. Si una aplicación incorporada es pobre o carece de funciones útiles, la comunidad de código abierto es más probable que construya una mejor. Un ejemplo es el construido en la aplicación de mensajería, que puede ser sustituido por Handcent SMS descargado del mercado de Android, la versión (3.0.4) ya que muestra el mensaje recibido en el dispositivo Android directamente, sin utilizar la barra de notificación.

La desventaja de esta libertad es que los usuarios tienen que tener mucho cuidado al desinstalar aplicaciones. Podrían fácilmente destruir partes de Android, por lo que no puede reiniciar. Otra desventaja de Android es su apertura, lo que lo hace más expuesto al malware (McAllister, n.d.).

ESTRUCTURA ANDROID, JAVA Y XML

Android es también la biblioteca de clases Java utilizada para crear aplicaciones para el entorno de software de Android. Java en Android hace uso de XML para declarar variables como cadenas y enteros. XML también se utiliza para controlar el diseño y el estilo de una aplicación (Ableson & Sen, 2009), (Conder & Darcey, 2009). Esta convención de capa independiente es similar a Hyper Text Markup Language (HTML) para contenido y Cascading Style Sheet (CSS) para estilos. Hay diferencias debido a la gran cantidad de idiomas y funcionalidades, ya que HTML no es un lenguaje de programación como lo es Java. Sin embargo, desde un nivel conceptual esta comparación puede hacerse. En lo que se refiere a aspectos prácticos, Java define la funcionalidad del botón, mientras que XML define el texto, el color, el tamaño de fuente y el tamaño de los botones.

VERSIONES DE ANDROID

Android viene en muchas formas. Desde el primer lanzamiento público abierto el 21 de octubre de 2008, Android utilizó la versión 1.1. Desde entonces, se encuentran versiones 1.5, 1.6, 2.0, 2.0.1 a la recientemente lanzada 7.1. NOUGAT

Desde el marco de referencia de los desarrolladores es importante considerar las versiones Android que serían soportadas por la aplicación desarrollada, además de garantizar el correcto funcionamiento para versiones antiguas y nuevas, otro enfoque es apuntar el desarrollo a cada versión de Android, pero este enfoque es costoso.

Para el desarrollo de aplicaciones Android se debe tomar en cuenta las necesidades de los interesados en términos de tiempo, financiamiento y lanzamiento. Para realizar el debug en varias versiones de Android, contamos con: Emuladores multi versión de GENYMOTION y también existe una utilidad web la cual brinda como servicio principal el correr la aplicación en las versiones que necesitamos.

ENTORNO DE DESARROLLO

Android SDK hace uso de lenguaje de programación Java, similar a Java Standard Edition (JSE), llamado Java Android Library. Esta es una ventaja para los desarrolladores familiarizados con los lenguajes de programación originados en la familia de lenguaje de programación C. La

sintaxis es igual a Java en términos de operadores, selecciones, iteraciones, manejo de archivos y más. Las clases y paquetes de Android más específicos utilizan otros nombres que no son similares a las ediciones de Java, como la clase de actividad y la clase de vista (Conder & Darcey, 2009).

Para desarrollar una aplicación de Android, los desarrolladores deben asegurarse de que el entorno de desarrollo tiene una versión Java 5 o superior. Hoy en día Java 8 es estable y no hay ninguna razón para no utilizarlo. Descargue e instale una versión de Java adecuada para el sistema operativo en desarrollo, ya que Java es independiente del sistema operativo, el desarrollador podría elegir libremente el sistema operativo del entorno de desarrollo.

ANDROID SDK

El Android SDK es mandatorio para los desarrolladores de Android, este contiene todos los paquetes, aplicaciones, marco de trabajo, librería de clases que el desarrollador necesita para crear aplicaciones de Android. El desarrollador debe descargar e instalar SDK luego configurar la ruta de las variables de entorno, esto es esencial para desarrollar en cualquier tipo de sistema de archivos.

ANDROID STUDIO

Android Studio es el entorno oficial de desarrollo (IDE) de Android, es un poderoso editor de código y herramientas de desarrollo. Dicho entorno ofrece más características que mejoran tu productividad cuando creas aplicaciones de Android (Google Inc., 2017).

Características como las siguientes:

- Un sistema de compilación flexible basado en Gradle
- Un emulador veloz con varias características
- Un entorno unificado donde puedes crear para todos los dispositivos
- Instalar, ejecutar y enviar los cambios a tu aplicación en ejecución sin crear una nueva aplicación
- Plantillas de código y la integración con GitHub que te ayudan a crear aplicaciones de características comunes e importar código de ejemplo
- Extensas herramientas de prueba y marcos de trabajo
- Herramientas de Lint para obtener rendimiento, usabilidad, compatibilidad de versiones y otros problemas
- Soporte integrado para la Plataforma Google Cloud, lo que facilita la integración de Google Cloud Messaging y App Engine.

DISPOSITIVO VIRTUAL DE ANDROID

Cada versión de Android tiene un dispositivo diferente (AVD) que necesita una previa configuración. La idea general es desarrollar la versión Android más baja posible, para que la aplicación sea compatible con más dispositivos móviles y por ende con más usuarios que deseen instalar la aplicación desarrollada. No todos los usuarios de Android actualizan la versión con frecuencia y hay algunos fabricantes de teléfonos que no admiten versiones nuevas sus dispositivos. La conclusión para el desarrollador es tomar decisiones informadas sobre las versiones de Android adecuadas, nunca usar una versión demasiado alta si no es necesario. El desarrollador debe tener en cuenta que una aplicación de Android en una versión inferior es compatible con una versión anterior (Conder & Darcey, 2009).

El AVD se puede configurar mediante la consola de administración de emuladores, herramienta disponible en Android Studio. Por conveniencia, se recomienda a los desarrolladores configurar todos los dispositivos virtuales posibles desde el principio. Este enfoque facilita el esfuerzo de desarrollo más adelante, cambiando de una versión de máquina virtual a otra, especialmente en propósitos de prueba. El ADV es un emulador que contiene el sistema operativo específico de Smartphone (Ableson & Sen, 2009). Esto es conveniente si el desarrollador no tiene un dispositivo propio. Sin embargo, el desarrollador debe tener cuidado de liberar una aplicación que no ha sido probada en un teléfono real.

VISTA DE PROYECTOS DE ANDROID

Un proyecto creado en Android Studio mantiene sus archivos ordenados y estructurados, aunque también existen algunos archivos escondidos de la vista. Los directorios principales los podemos encontrar en la parte izquierda del explorador del IDE Android Studio. Y los directorios y archivos ocultos se los puede encontrar con el explorador de Windows en la carpeta del proyecto, en el caso de utilizar una MAC lo podemos hacer con el FINDER. La estructura está compuesta de los siguientes directorios

module-name/

 build/

 en este directorio podemos visualizar archivos de resultados de compilación.

Libs/

Contiene bibliotecas privadas.

src/

Son los archivos de recursos de proyecto Android:

androidTest/

Este directorio tiene en su interior archivos de código para las pruebas de instrumentación que se corren en un dispositivo Android.

main/

son los archivos de conjunto de origen “principales”: los recursos y código de Android compartidos por todas las opciones de compilación (los archivos para otras opciones de compilación residen en directorios del mismo nivel.

AndroidManifest.xml

Es el archivo de descripción principal de aplicación y cada uno de sus partes. Para descubrir más información podemos consultar la documentación de AndroidManifest.xml

java/

Son los archivos con código fuente Java.

jni/

Son los archivos de código nativo en el cual se usa la interfaz de Java (JNI).

gen/

Son el resultado de archivos Java generados por Android Studio

res/

en este directorio están los recursos de aplicación, como archivos de elementos de diseño, archivos de diseño y strings de la interfaz de usuario.

assets/

En el directorio assets tenemos los archivos que se debe compilar para genera un archivo. Apk tal como está. Se puede explorar el directorio del mismo modo que un sistema de archivo común usando URI y leer archivos como transmisiones de bytes usando el AssetManager. Por ejemplo, es la mejor opción para la ubicación de texturas y datos de juegos.

test/

Archivos de código para realizar de forma local las pruebas que se ejecutan en tu máquina Virtual Java.

build.gradle (módulo)

Este archivo describe las configuraciones de compilación específicas para el módulo.

build.gradle (proyecto)

El archivo define la configuración de compilación que se ejecuta a todos los módulos. Este es parte primordial del proyecto. Por lo tanto, se debe someterlos a control de revisión con todos los códigos fuente generados restantes (Google Inc., 2017).

SERVICIO WEB

Según (Garcia & Munilla, 2004) “Es una tecnología de integración basada en un conjunto de estándares de internet que permite la comunicación y ejecución de transacciones de distintas aplicaciones sin la complejidad de sus predecesoras”.

El servicio web es un componente o software que reside en la web con una interfaz que admite las solicitudes y devuelve información en formatos estandarizados al cliente. Los servicios web no están ligados directamente con una aplicación en particular. Un servicio web posee un modelo similar al cliente – servidor, carece de estados de sección y se basan en estándares de comunicación como los formatos XML, Json. Los servicios web también podemos definir como formatos estandarizados y abiertos (XML, SOAP, http) basados en la web para aplicaciones que interrelacionan una con otra con el propósito de intercambiar información.

“Los Web Services (WS) ofrece una un significado estándar para inter operar entre diferentes aplicaciones de software corriendo en diferentes plataformas y/o marcos de trabajo. El W3C pretende diseñar la arquitectura, definirla y crear el núcleo de tecnologías que hagan posible los servicios Web” (<http://www.w3.org/2002/ws/desc/>, 2011).

Los servicios más usados son los SOAP Y REST.

SOAP: (Simple Object Access Protocol)

Los servicios web se basan en XML, la mayoría de ellos se adhieren al estilo RPC de diseño de aplicaciones (llamando a métodos remotos en un servidor y obteniendo una respuesta) y usan tres pilares principales:

- WSDL - Lenguaje de descripción de servicio web - se utiliza para describir un servicio en términos de operaciones disponibles, parámetros, etc.
- SOAP - Simple Object Access Protocol - utilizado para construir mensajes de interacción entre las entidades involucradas (cliente, servidor).
- UDDI - Descripción, descubrimiento e integración universal - se utiliza para clasificar y publicar los servicios web disponibles en un repositorio y permitir el descubrimiento por los usuarios potenciales.

Los SOAP Web Services tienden a tener una sobrecarga alta y generalmente tienen mensajes muy detallados, pero pueden ser buenos si necesita implementar funcionalidades e interacciones más complejas en su aplicación.

REST: (Representational State Transfer) es un modelo de arquitectura que especifica restricciones, como una interfaz uniforme que, si se aplica a un servicio, induce propiedades deseables, como rendimiento, escalabilidad y modificabilidad que posibilita servicios a trabajar en la mejor forma. En el estilo de arquitectura REST, datos y funcionalidad son considerados recursos a los que se accede usando (URIs) enlaces típicos en la internet. Los recursos son soportados con el uso de operaciones simples bien definidas. El estilo de arquitectura REST restringe una arquitectura cliente servidor y está diseñada para usar un protocolo de comunicación sin estado. En el estilo de arquitectura REST, Clientes y servidores intercambian representaciones de recursos con el uso de una interfaz y protocolo estandarizado.

Descripción de los servicios

“Web Services Description Language (WSDL): Lenguaje de Descripción de los Servicios Web. Se trata de un lenguaje para describir Servicios Web. La especificación define el lenguaje básico que puede usarse para describir servicios web basados en un modelo abstracto de lo que ofrece el servicio. También define los criterios de conformidad de los documentos en relación a este

lenguaje” afirmación consultada de (<http://www.w3.org/TR/wsdl20/>, [Consulta: 13 de Febrero del 2017]).

Podemos definir que “Web Services Choreography Description Language (WS-CDL): Lenguaje de Descripción de la Coreografía de los Servicios Web. Es un lenguaje basado en XML que describe colaboraciones peer to peer de los participantes definiendo, desde un punto de vista global, un comportamiento observable común y complementario; donde ordenado el mensaje, intercambia el resultado de acuerdo a un objetivo de negocios común” según (<http://www.w3.org/TR/ws-cdl-10/>, [Consulta: 13 de Febrero del 2017]).

Según (<http://www.google.com/apis/>, [Consulta: 13 de Febrero del 2017]) “Los servicios web que se basan en XML permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado dichas aplicaciones e independientemente del sistema operativo o plataforma en que se ejecuten y de los dispositivos utilizados en el acceso. Los servicios Web XML, aunque sean independientes entre sí, pueden vincularse para realizar una tarea. Por ejemplo, Google, utiliza un servicio Web-Google Web APIs- basado en los estándares SOAP y WSDL que permite programar en Java, Perl o Visual Studio.NET y que sirve para la recuperación de información permitiendo utilizar este buscador en distintas plataformas y Servicios Web.

Por su parte, la función del lenguaje WSDL (Web Service Description Language) es decirle a una aplicación qué formato usar para comunicarse, especificando por medio de un lenguaje estándar, tanto la dirección del servicio como la interfaz que se va a utilizar. WSDL es un lenguaje basado en XML para describir servicios en la Web. Ofrece a los proveedores de servicios, un formato básico de descripción de las peticiones de servicios web sobre diferentes protocolos o codificaciones”.

Existe un grupo de trabajo dentro del W3C, el Web Services Description Working Group (<http://www.w3.org/2002/ws/desc/>, [Consulta: 13 de Febrero del 2017]). Define que “Que analiza y desarrolla el lenguaje WSDL. WSDL se usa para describir qué puede hacer un servicio web, dónde reside, y cómo invocarlo. WSDL define los servicios como colecciones de puntos finales

de la red o puertos. En WSDL la definición abstracta de puntos finales y mensajes se separa de su concreto despliegue en la red o formato de datos ligados. Esto permite reutilizar las definiciones abstractas de los mensajes, que son descripciones abstractas de los datos que están siendo intercambiados, y los tipos de puerto, que son colecciones abstractas de operaciones. El protocolo concreto y las especificaciones del formato de datos para un tipo particular de puerto constituye un enlace reutilizable. Un puerto se define por asociación a una dirección de red con un enlace reutilizable; una colección de puertos define un servicio”. Y, así, un documento WSDL usa los siguientes elementos en la definición de servicios en red:

Tipos

Un contenedor para definiciones del tipo de datos que usan algunos tipos de sistemas (tal como XSD).

Mensaje

Una definición abstracta tipo del dato que está siendo comunicado.

Operación

Una descripción abstracta de una acción soportada por el servicio.

Tipo de puerto

Un conjunto abstracto de operaciones soportadas por uno o más puntos finales.

Conexión

Un protocolo concreto y una especificación de formato de datos para un tipo de puerto particular.

Puerto

Un punto final individual definido como una combinación de una conexión y una dirección de la red.

Servicio

Una colección de puntos finales relacionados.

IOS

Apple iOS es un sistema operativo móvil propietario que se ejecuta en el iPhone, iPad y iPod Touch. Apple iOS se basa en el sistema operativo Mac OS X para equipos de sobremesa y portátiles. El kit para desarrolladores de iOS proporciona herramientas que permiten el desarrollo

de aplicaciones iOS. Diseñado para su uso con los dispositivos multitáctiles de Apple, el sistema operativo móvil admite la entrada a través de la manipulación directa. El sistema responde a varios gestos del usuario, como pellizcar, tocar y deslizar.

Cuota de mercado

A partir de noviembre de 2016, Apple iOS su cuota de dispositivos en el mercado fue del 12,5% en todo el mundo, por lo que es el segundo SO más popular móvil detrás de Google Android, según IDC (International Data Corporation) empresa estadounidense de investigación, análisis y consultoría de mercados, especializada en tecnología de la información, telecomunicaciones y tecnología de consumo, desarrollo de software.

Apple iOS incluye las siguientes características:

- Wi-Fi, Bluetooth y conectividad celular, junto con soporte de VPN;
Soporte de búsqueda integrado, que permite la búsqueda simultánea a través de archivos, medios, aplicaciones y correo electrónico;
- Soporte de reconocimiento de gestos - por ejemplo, sacudir el dispositivo para deshacer la acción más reciente;
- Push email;
- Safari navegador móvil;
- Cámaras frontales y traseras integradas con capacidades de vídeo;
- Acceso directo a la App Store de Apple y al catálogo iTunes de música, podcasts, programas de televisión y películas disponibles para alquilar o comprar;
- Compatibilidad con el servicio cloud de Apple, iCloud;
- Siri asistente personal;
- Comunicaciones multiplataforma entre dispositivos Apple a través de AirDrop; y
- Apple Pay, que almacena los datos de la tarjeta de crédito de los usuarios y les permite pagar por bienes y servicios directamente con un dispositivo iOS.

Apple iOS era originalmente conocido como iPhone OS. La compañía lanzó tres versiones del sistema operativo móvil con ese nombre antes de que el iOS 4 debutara en junio de 2010.

El 12 de octubre de 2011, Apple lanzó iOS 5, que amplió el número de aplicaciones disponibles a más de 500.000. Esta versión de iOS también agregó el Centro de Notificación, una aplicación de cámara, Siri y más. Presentado el 11 de junio de 2012, iOS 6 incluía una aplicación de Mapas y la aplicación de programa de fidelización y almacenamiento de boletos.

Lanzado el 18 de septiembre de 2013, iOS 7 presentó una interfaz de usuario completamente rediseñada. En septiembre de 2014, iOS 8 introdujo Continuity, un sistema multiplataforma que permite a los usuarios de múltiples dispositivos de Apple para recoger en uno donde dejaron de otro. Otras nuevas características incluyeron la aplicación Fotos y Apple Music.

Apple iOS 9 e iOS 10 - lanzados, respectivamente, el 16 de septiembre de 2015 y el 13 de septiembre de 2016 - incluyeron actualizaciones como una sección de notificaciones mejorada, capacidades de iMessage mejoradas e integración de Siri con aplicaciones de terceros (techtargat, 2017).

XCODE

“Xcode se introdujo el 24 de octubre de 2003 junto con la versión 10.3 de Mac OS X, siendo desarrollado a partir del anterior entorno de desarrollo, Project Builder, al que sustituyó. Project Builder, a su vez, también era una herencia de la compañía NeXT, fusionada con Apple en 1996. La aparición de Xcode 2.1 en junio de 2005 fue significativa porque proporcionó a la comunidad de desarrolladores las herramientas para crear binarios universales que permiten al software creado para Mac OS X ser ejecutado tanto en la arquitectura PowerPC como en la nueva, basada en Intel (x86). Esta versión integró además las herramientas y marcos de trabajo Web Objects de Apple para construir aplicaciones y servicios web de Java, que anteriormente se vendían como un producto separado” (wikipedia, 2017).

SWIFT

Swift es un lenguaje de programación moderno, fácil de usar y potente iniciado por Apple, y hace que sea mucho más fácil para los desarrolladores crear y entregar experiencias de aplicaciones excelentes. Lanzado a open source en diciembre de 2015, el atractivo de Swift está creciendo rápidamente, con los esfuerzos comunitarios ahora en marcha para llevar el idioma a nuevas plataformas y utilizarlo para algo más que el desarrollo de aplicaciones iOS.

Swift es rápido, seguro desde el principio, requiere mucho menos código y es rápidamente utilizable incluso por los programadores principiantes. Con Swift, aún tendrás los conceptos de programación comunes a todos los lenguajes de desarrollo, incluyendo variables, bucles,

estructuras y arreglos, pero estos elementos se exponen de una manera más intuitiva que hace que sea fácil ver y entender lo que está sucediendo en tu aplicación.

JAVA

“Java es un lenguaje de programación y plataforma de computación lanzado por primera vez por Sun Microsystems en 1995. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado, y más se crean todos los días. Java es rápido, seguro y confiable. Desde computadoras portátiles a centros de datos, consolas de videojuegos, supercomputadoras científicas, teléfonos celulares a Internet, ¡Java está en todas partes!” (Oracle Corporation, 2017).

XML

XML, creado por el World Wide Web Consortium (W3C), forma parte de una familia numerosa de lenguajes de marcado y se define como un metalenguaje - un lenguaje que describe otros lenguajes. Uno de los objetivos del W3C era hacer que XML fuera "sin opción" para que permaneciera puro, a diferencia del HTML que tiene muchas convenciones diferentes y como resultado se procesa de manera diferente en varios navegadores, lo que dificulta la presentación de datos de manera uniforme.

A partir de las pruebas de marzo de 2005, publicado por el W3C, se reveló que Microsoft Internet Explorer 6.0 SP2 tenía capacidad limitada en XML, según se informa utilizar su propio sabor del lenguaje que puede no siempre cumplir con los estándares establecidos. Netscape tenía buena compatibilidad con algunos problemas en la versión beta 8.0, mientras que Firefox y Mozilla tuvieron los mejores resultados de los navegadores libres, con una capacidad de procesamiento de XML totalmente implementada y 100% compatible en todas sus versiones de navegador disponibles hasta la fecha.

Debido a que las páginas XML proporcionan mucha más flexibilidad que las páginas HTML, se espera que reemplace a HTML como el idioma elegido. Para obtener más información, puede visitar el sitio oficial del W3C. Tutoriales en línea y muchos libros también están disponibles. Aprender el idioma puede requerir un aumento, pero los expertos predicen que la inversión vale la pena

JSON

JSON (JavaScript Object Notation) es un formato de almacenamiento de datos legible por máquina que puede incluir información. Douglas Crockford originalmente especificó y popularizó la estructura alrededor de 2001. Su propósito ha sido comparado con XML, sin embargo, JSON requiere menos espacio de almacenamiento y es más legible que la especificación XML más tradicional.

La notación JSON es muy similar a la estructura de lenguaje Javascript de arrays y objetos asociativos. Consiste en pares de valor de nombre existentes dentro de una relación padre-hijo. La adopción de JSON ha crecido rápidamente y es utilizada por un número creciente de sitios web en todo el mundo. AJAX (un método común de comunicación entre sitios web y servidores) a menudo utiliza JSON como el lenguaje de comunicación.

PHP

Personal Home Page (PHP) es un lenguaje de servidor, que se utiliza como lenguaje de programación para desarrollar sitios web. Rasmus Lerdorf diseñó PHP en 1994 pero fue desarrollado por Zend Technologies. Su código está incrustado en el código HTML para su ejecución. También se utiliza con Web Content Management Systems y Web Frameworks. El código escrito en PHP es interpretado por un intérprete PHP, que se implementa en el servidor.

PRINCIPALES USOS

Es un lenguaje de script de uso general, basado en servidor, que se ejecuta en un servidor web. El principal uso de PHP es desarrollar páginas web dinámicas que cargan contenido, imágenes y otros datos de las bases de datos basadas en la función escrita en el archivo PHP. El código escrito en un archivo PHP es ejecutado por el tiempo de ejecución de PHP. Aparte de ser utilizado como un lenguaje de scripting del lado del servidor, es un lenguaje de programación web bien escrito, que junto con CSS y HTML ha estado cambiando el Internet. PHP también se utiliza para scripts de línea de comandos. Por lo general, la salida del código PHP es HTML, pero no es una regla dura y rápida, ya que su salida puede ser JSON, XML o datos binarios también.

MYSQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporación y está considerada como la base datos de fuente abierta más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL A.B. fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.

SQLITE

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kiB) biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB.

SECCIÓN III

METODOLOGÍA

Para el presente trabajo de investigación utilizamos dos métodos, uno de investigación y uno para el desarrollo, que nos servirán como guía para cumplir satisfactoriamente y concluir de manera satisfactoria.

METODOLOGÍA DE INVESTIGACIÓN

Para realizar el trabajo de investigación vamos a utilizar el método de investigación deductivo, cualitativo y exploratorio.

Método Deductivo: Es un método científico que va de lo general a lo particular ya que pretende desarrollar una teoría empezando por formular puntos de partida y deduciendo luego sus consecuencias (Chung, 2008a). El método deductivo parte de datos generales aceptados como valederos, para deducir a través del razonamiento lógico y estadístico varias suposiciones, es decir, se partirá de un argumento o de la formulación de una hipótesis, en donde para comprobar su validez se compara con la información existente (Chung, 2008b).

Para la presente tesis se requiere una recopilación importante de información con datos reales, aceptados como valederos, sobre evaluación de usabilidad de aplicaciones multiplataforma en dispositivos móviles, para que a través de un razonamiento lógico se logre generar un reporte con respecto a la información existente.

Investigación Cualitativa: Tiene como principal objetivo la recopilación y descripción de las cualidades de un fenómeno a partir de indagaciones (Chung, 2008), en el caso de la presente tesis este método es necesario, porque que se va a realizar una descripción y un estudio minucioso de los datos encontrados sobre la evaluación de usabilidad de aplicaciones multiplataforma en dispositivos móviles.

Investigación exploratoria: dicha Investigación se realiza sobre un problema o situación que proporcione información al investigador. La investigación está destinada a proporcionar detalles donde existe una pequeña cantidad de información. Puede utilizar una variedad de métodos tales

como estudios de ensayo, entrevistas, discusiones en grupo, experimentos u otras tácticas con el propósito de obtener información (Business Dictionary, 2017).

Para la presente investigación se requiere diagnosticar la situación del uso de técnicas de evaluación de usabilidad de aplicaciones multiplataforma en dispositivos móviles.

OBJETO DE ESTUDIO

Tomando en cuenta que el objeto de estudio se centra en la implementación de un software para teléfonos inteligentes, que permite acceder a los servicios publicados en el portal web de la Universidad Tecnológica Israel.

ANÁLISIS E INTERPRETACIÓN DE LA ENCUESTA

Para el presente trabajo se aplicó la investigación cualitativa usando como herramienta principal la encuesta a 268 alumnos, la población es tomada en cuenta de acuerdo a la cantidad de alumnos que sufragaron en las últimas elecciones internas de la institución 884, con estos datos la fórmula planteada para este caso es la siguiente.

$$n = \frac{\sigma^2 * N * p * q}{E^2(N - 1) + \sigma^2 * p * q}$$

Donde:

$\sigma = 1.96$ (nivel de confianza si la seguridad es del 95%)

$N =$ población (884)

$n =$ Muestra

$p =$ Proporción esperada (0.5, probabilidad de ocurrencia)

$q = 1 - p$ ($1 - 0.5 = 0.5$, probabilidad de no ocurrencia)

$E =$ precisión (5%).

$$n = \frac{1.96^2 * 884 * 0.5 * 0.5}{0.05^2(884 - 1) + 1.96^2 * 0.5 * 0.5}$$
$$n = 267.99$$

Para la realización de la encuesta se utilizaron herramientas disponibles en la web. En este caso utilizamos Google Forms la dirección es la siguiente <https://goo.gl/forms/OJJxFsgfl9tEWFoA2>.

Y para la difusión de esta encuesta se utilizó varios medios que se detalla a continuación:

- Redes sociales
- Correo electrónico
- Portal web de la Universidad Tecnológica Israel

Los resultados de la encuesta lo detallamos a continuación:

Posee un teléfono inteligente (SmartPhone)

271 responses

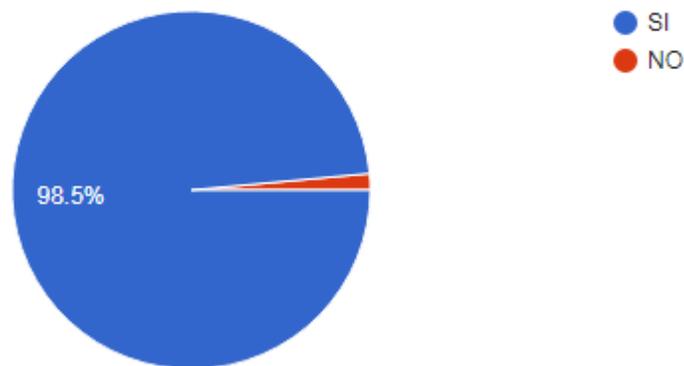


Figura 1 Resultado encuesta 1

Fuente: Elaborador por los autores

En la pregunta realizada se evidencia que en forma mayoritaria los alumnos poseen un teléfono inteligente, por lo que el proyecto es viable.

Que sistema operativo tiene su SmartPhone

268 responses

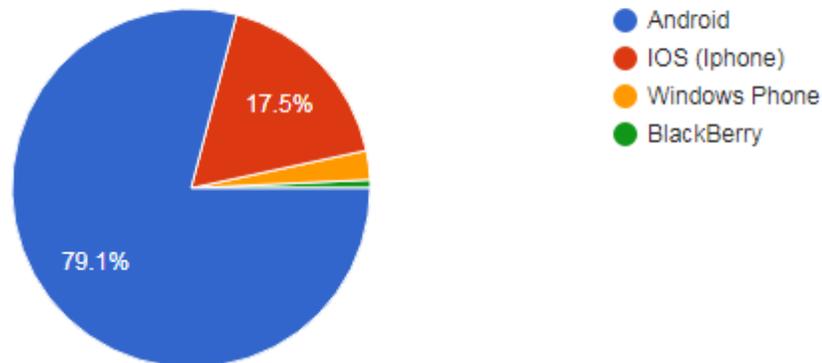


Figura 2 Resultado encuesta 2

Fuente: Elaborador por los autores

En la pregunta realizada se evidencia que los sistemas operativos más utilizados son Android e IOS, estos indicadores nos enfocan a desarrollar las aplicaciones para ambas plataformas y de esta manera abarcar a la mayoría de estudiantes.

La Universidad Israel debería disponer de una aplicación móvil?

271 responses

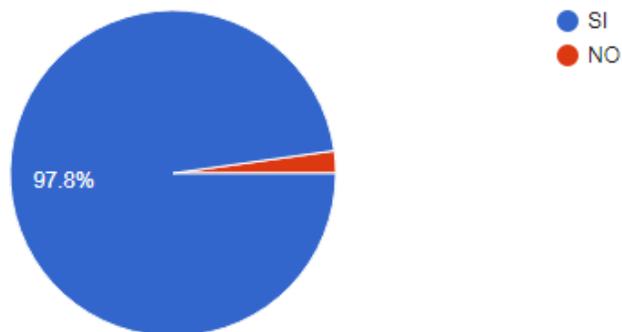


Figura 3 Resultado encuesta 3

Fuente: Elaborador por los autores

La mayoría de los alumnos encuestados considera que debe existir una aplicación móvil de la Universidad Tecnológica Israel, dando la razón suficiente para que el proyecto se realice.

Que secciones debería tener la aplicación móvil?

265 respuestas

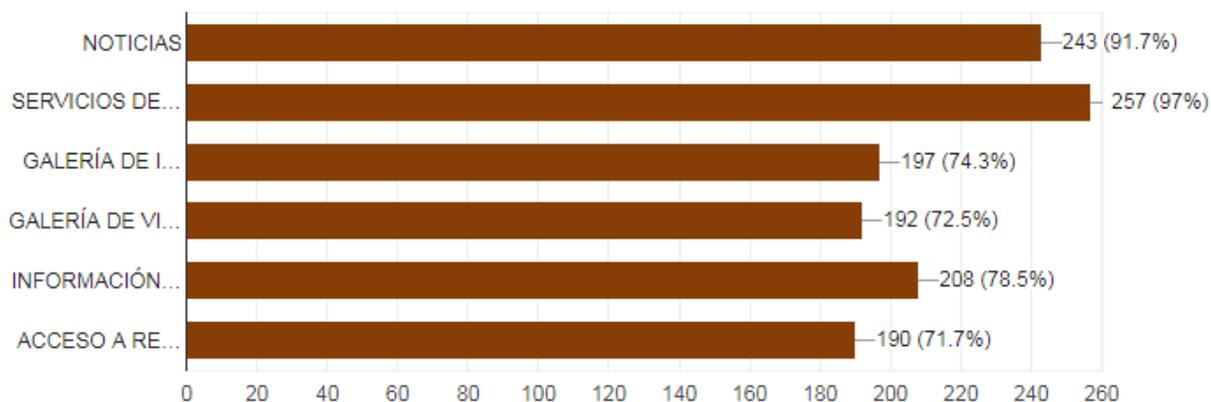


Figura 4 Resultado encuesta 4

Fuente: Elaborador por los autores

De los indicadores obtenidos en la pregunta podemos concluir que todas las secciones propuestas deben ser desarrolladas ya que los estudiantes en su gran mayoría las seleccionaron.

Le interesaría leer noticias de otros medios para la comunidad universitaria?

265 respuestas

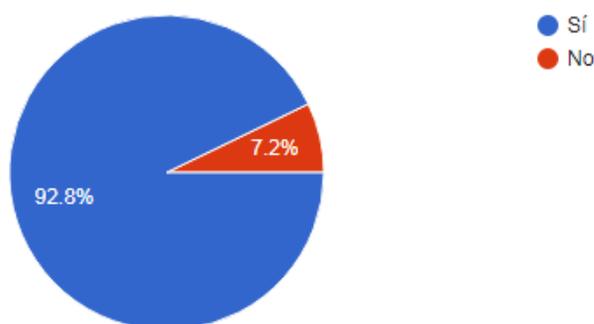


Figura 5 Resultado encuesta 5

Fuente: Elaborador por los autores

Los alumnos consideran que en la sección de noticias se incluya la información que genere otra fuente de interés para la comunidad universitaria.

Con que frecuencia utiliza los servicios universitarios del portal web de la Uisrael?

256 respuestas

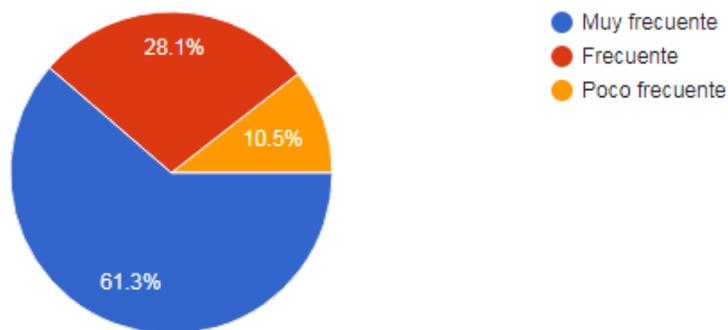


Figura 6 Resultado encuesta 6

Fuente: Elaborador por los autores

Se evidencia que los servicios universitarios publicados en el portal web, son utilizados frecuentemente por los alumnos.

En caso de contar con una app, su ingreso a los servicios estudiantiles de la Uisrael sería más frecuente?

256 respuestas

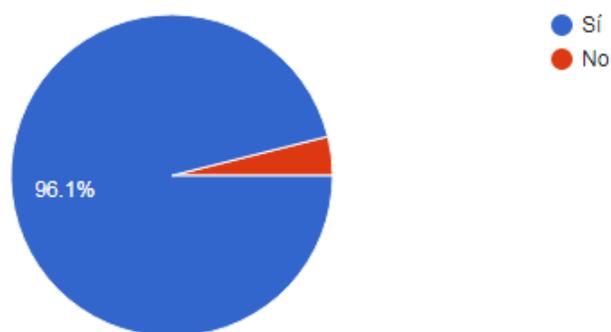


Figura 7 Resultado encuesta 7

Fuente: Elaborador por los autores

Los alumnos encuestados aprueban que la aplicación móvil facilitará el acceso a los servicios estudiantiles y como resultado su uso será más frecuente.

Que servicios del portal web debería tener la aplicación móvil?

265 respuestas

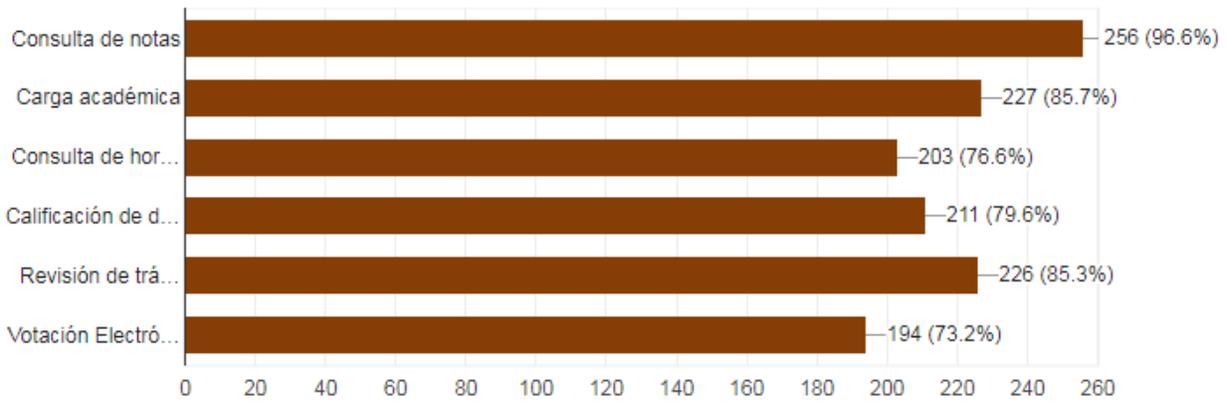


Figura 8 Resultado encuesta 8

Fuente: Elaborador por los autores

Los servicios publicados en el portal web fueron aceptados por los alumnos para que sean parte de la sección servicios estudiantiles de la aplicación móvil.

Cree que el envío de notificaciones mejorará la comunicación con los estudiantes?

266 respuestas

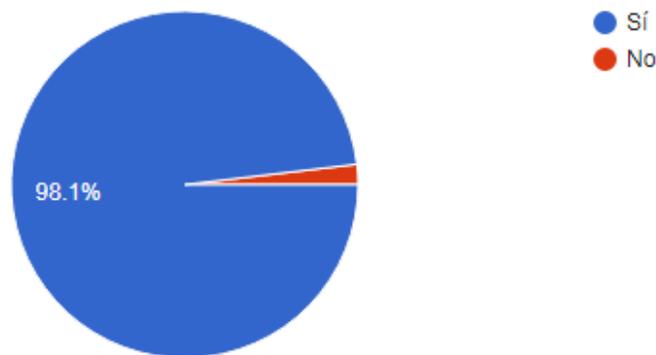


Figura 9 Resultado encuesta 9

Fuente: Elaborador por los autores

Con los datos obtenidos de esta pregunta, se confirma que las aplicaciones móviles deben implementar la utilidad de notificaciones para mejorar la comunicación con los alumnos.

DIAGNOSTICO

Con la información obtenida de la encuesta realizada podemos concluir que los alumnos ven con agrado y aprueban la creación de una aplicación móvil. Además, podemos definir cuáles serán los principales servicios que debe disponer la aplicación.

METODOLOGÍA DE DESARROLLO

De acuerdo a (Andersin, Niemi, & Hirvonen, 2005), el proceso Mobile-D debe ser usado por un equipo pequeño de desarrolladores, trabajando hacia la entrega de un producto dentro de aproximadamente 10 semanas. Los principales elementos envueltos en diferentes prácticas a través de los diferentes ciclos del desarrollo.

- Fase y colocación
- Línea de arquitectura
- Pruebas de desarrollo móvil
- Integración continua
- Emparejar programación
- Métricas
- Procesos de mejora de software ágil
- Cliente fuera de lugar
- Enfocado al usuario central

Mobile-D apunta hacia la entrega del producto dentro de aproximadamente 10 semanas. ¿Cómo consigue el método esto? Se basa en prácticas ágiles de desarrollo, las cuales se centran en confiar en el juicio del equipo, en la comunicación en los clientes, en la rápida entrega de los productos en las frecuentes pruebas durante el ciclo de vida del proyecto.

La comunicación es crucial dentro del equipo y verticalmente en toda la organización. Siendo una de las tareas más difíciles, la comunicación a menudo determina el éxito del proyecto en general.

La interacción continua y la comunicación con los clientes permite a los directores de proyectos discernir información inestimable sobre el mercado objetivo y sus necesidades. Esta área presenta desafíos únicos ya que la base de clientes no siempre está dispuesta a cooperar, comunicar o proporcionar información relevante.

El proceso Mobile-D consta de 5 fases:

Explorar

En la primera fase, una estrategia y los componentes del proyecto son determinados por el equipo de desarrollo y se divide en tres etapas:

- Establecimiento de las partes interesadas
- Definición del alcance
- Establecimiento del proyecto

El equipo completará las tareas específicas de esta fase, como determinar que clientes tomaran un papel operativo de le proceso de desarrollo.

Inicializar

El equipo se reúne con las partes interesadas activas para comprender el producto y ensamblar activos cruciales, tales como comunicaciones, recursos tecnológicos y físicos, para comenzar con las actividades de producción. Las tres etapas de esta fase son:

- Preparación del proyecto
- Planificación inicial
- Día de prueba

Producción

La mayoría de la implementación tiene lugar y debe completarse durante esta fase. Los días de planificación, los días laborables y días de lanzamiento son la fuerza motriz estructural detrás de esta fase.

- Días de planificación: “Los días de planificación están destinados a mejorar el proceso de desarrollo, priorizar y analizar los requisitos, planificar el contenido de iteración y crear pruebas aceptables que se ejecutaran más tarde en los días de lanzamiento” (Spataru, 2010).
- Días laborables: durante los días laborables, las funcionalidades se implementan utilizando la práctica de desarrollo controlado por pruebas.
- Días de lanzamiento: los días de lanzamiento se gasta creando un prototipo de trabajo que será certificado a través de pruebas de aceptación.

Estabilizar

Durante la fase de estabilización, se produce la finalización del producto. Esto puede abarcar ajustes o modificaciones, o en un proyecto de varios equipos integrando subsistemas si es necesario.

Pruebas y arreglo del sistema

Como su nombre lo indica, esta fase implica la prueba del producto y los lazos estrechamente con la cuarta fase, la estabilización. La estabilización, la prueba y la fijación del sistema puede continuar con el ciclo hasta que el producto cumpla con las expectativas y se cumplan los requisitos del cliente. Hay una importancia crucial en la cohesión del equipo, la armonía y la confianza mutua. Debido a que se requiere un equipo muy unido y comprometido, lo equipos pequeños no más de 10 miembros son adecuados para Mobile-D.

PROPUESTA

Para el desarrollo del presente trabajo se usa las fases del Mobile-D, para realizar la aplicación móvil y plataforma web del cual se consume los servicios, para dar solución a los objetivos planteados.

Exploración

Establecimiento de las partes interesadas

La aplicación tiene dos tipos de usuario

- Establecimiento del cliente

Tabla 1 Partes Interesadas

NOMBRE	DESCRIPCIÓN	ASOCIADOS
Usuario final	Son los usuarios que poseen la aplicación instalada en su teléfono inteligente	Usuario

Administrador	Es el encargado de ingresar las noticias, servicios, notificaciones y galería en la consola de administración	Administrador
---------------	---	---------------

Fuente: Elaborado por los autores

Definición del alcance

Colección de requerimientos funcionales

- Envío de notificaciones a los usuarios
- Consulta de noticias de la universidad
- Consumo de servicios publicados en la web de la universidad
- Ingreso a la galería
- Información general de la universidad
- Contactos

Colección de requerimientos no funcionales

- Todos los procedimientos están enfocados en la consulta de información disponible en el portal web y no se puede realizar el ingreso de información.

Establecimiento del proyecto

Selección del ambiente

Para el desarrollo del proyecto es necesario:

- Equipo en la que se implementara el servidor web.
- Conexión a internet
- Teléfonos inteligentes con sistemas operativos IOS y Android
- Plan de datos para los Smartphone para realizar las pruebas de funcionamiento
- Equipos para el desarrollo de Android e IOS

Asignación del personal

Tabla 2 Asignación del personal

Desarrolladores	
Marcelo Xavier Secaira Larco	<ul style="list-style-type: none"> • Programación de Android • Programación consola web
Diego Román Noboa	<ul style="list-style-type: none"> • Programación IOS • Programación consola web

Fuente: Elaborado por los autores

Definición de la línea de arquitectura

La arquitectura está basada en la tecnología de servicios web REST. La cual nos permitirá interactuar con la con los servicios disponibles en el portal web de la Universidad Tecnológica Israel.

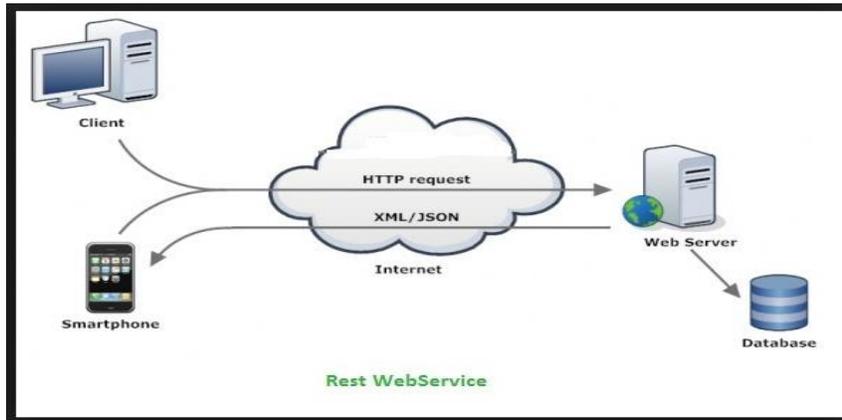


Figura 10 Arquitectura de Servicios Web tipo REST

Fuente: Elaborador por el autor

Establecimiento de procesos

Tabla 3 Lista de verificación

Lista de verificación del plan del proyecto			
Fase de exploración			
Requerimientos iniciales	SI	NO	N/A
Todos los requerimientos iniciales funcionales han sido incluidos en el plan del proyecto	X		
Todos los requerimientos no funcionales han sido incluidos en el plan del proyecto	X		
Calendario y ritmo			
El calendario general ha sido incluido en el plan del proyecto	X		
El ritmo planeado ha sido definido en el plan del proyecto	X		
Recursos	SI	NO	N/A

El plan del proyecto se ha actualizado con los grupos de interés identificados y sus miembros	X		
Se ha actualizado el plan del proyecto de la información relativa al software seleccionado, herramientas de desarrolla, terminales, etc.	X		
El plan del proyecto se ha actualizado con los miembros del equipo del proyecto identificado	X		
Entrenamiento	SI	NO	N/A
Las necesidades de capacitación del equipo del proyecto se han incluido en el plan del proyecto	X		
El calendario de capacitación ha sido incluido en el plan del proyecto			X
Documentación	SI	NO	N/A
Los documentos a ser producidos en el proyecto han sido incluidos en el plan del proyecto (arquitectura, interfaz de usuario)	X		
La duración de vida de cada documento ha sido incluida en el plan del proyecto			X
Aseguramiento de calidad	SI	NO	N/A
Los procedimientos de garantía de calidad se han definido en el plan del proyecto (documentación, código y producto)	X		
Iteración			
Iteración	SI	NO	N/A
El plan del proyecto se ha actualizado con respecto a los requisitos de prueba seleccionados para la iteración	X		
El plan del proyecto se ha actualizado con respecto a la realización del ensayo seleccionado requisitos para la iteración	X		
La definición de la línea de arquitectura se ha incluido en el plan del proyecto	X		

Fuente: Elaborado por los autores

Inicialización



Figura 11 Inicialización del proyecto

Fuente: Elaborado por los autores

Preparación del proyecto



Figura 12 Preparación del proyecto

Fuente: Elaborado por los autores

Recursos físicos

Tabla 4 Recursos Físicos

Recurso	Utilidad
Teléfono con sistema operativo Android	Pruebas de funcionamiento
Teléfono con sistema operativo IOS	Pruebas de funcionamiento
Ordenador	Desarrollo de la aplicaciones y pruebas de funcionamiento del web Service

Fuente: Elaborado por los autores

Recursos técnicos

Tabla 5 Recursos Técnicos

Herramientas	Utilidad
Android estudio	IDE de desarrollo para Android
Xcode	IDE de desarrollo para IOS
Netbeans	IDE de desarrollo para la consola web
Base de datos	Utilidad
Mysql	Base de datos que utilizamos en la consola de administración
Sqlite	Base de datos utilizada en dispositivos móviles
Lenguajes de programación	Utilidad
Java	Lenguaje de programación utilizado para Android
php	Lenguaje de programación utilizado para a consola web
Swift	Lenguaje de programación utilizado para sistema operativo IOS

Fuente: Elaborado por los autores

Entrenamiento

Para el presente trabajo de investigación se utiliza conocimientos previos en la utilización de herramientas de desarrollo como Android estudio para Android y Xcode para IOS. Además, son necesarios conocimientos en la implementación de servicios web con arquitectura REST.

Planificación día inicial

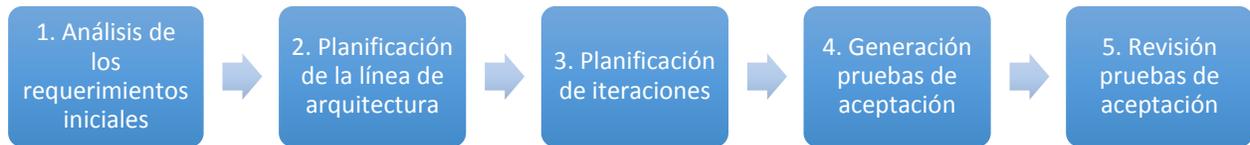


Figura 13 Planificación del día inicial

Fuente: Elaborado por los autores

Análisis de requerimientos iniciales

- Información que estarán disponibles en el web servicie
- Creación de una aplicación móvil para dispositivos con sistema operativo Android
- Creación de una aplicación móvil para dispositivos con sistema operativo IOS
- Creación de una consola de administración web que permitirá la administración de los servicios web disponibles para los Smartphone.

Planificación de la línea de arquitectura

El proyecto contara con una arquitectura basada en servicios web REST.

Para la utilización de la tecnología REST debemos tener en cuenta una serie de diseños básicos:

- Es una forma de comunicación por medio de un protocolo cliente / servidor sin estado. Básicamente es una cadena de texto que se transmite basada en el protocolo HTTP y no almacenamos la sección en ningún lado.
- El método REST tiene las operaciones bien definidas
 - INSERTAR POST
 - SELECCIONAR GET
 - ACTUALIZAR PUT
 - BORRAR DELETE
- La sintaxis es universal para la utilización de los recursos disponibles a través de la URI.
- El formato de la información es únicamente HTML, XML o JSON.

En el proyecto planteado los clientes son los dispositivos móviles y el servidor es nuestra consola de administración WEB.

El formato a utilizar es JSON.

Las solicitudes de información son solo consulta tipo GET. No se utiliza los otros métodos para garantizar la seguridad de nuestra información.

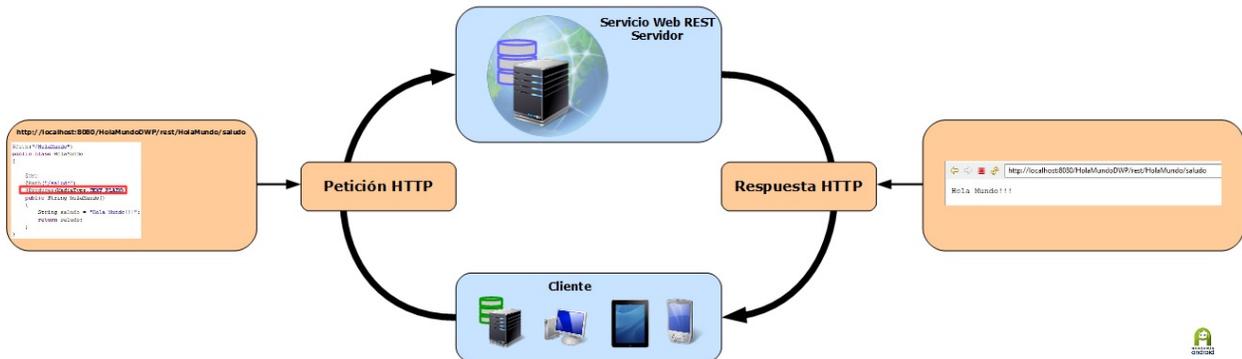


Figura 14 Arquitectura Base

Fuente: Elaborado por los autores

Modelo Relacional

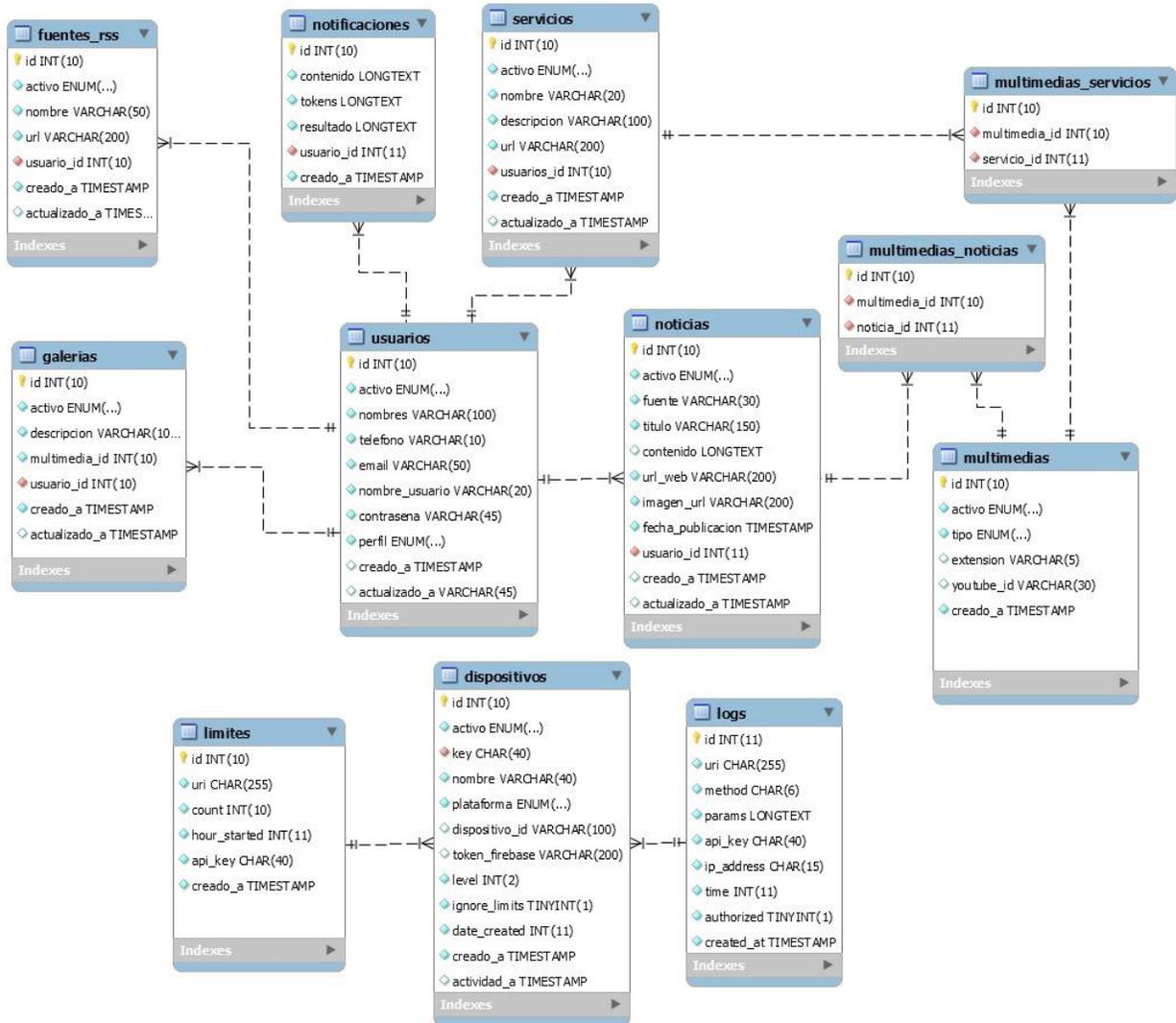


Figura 15 Modelo Relacional

Fuente: Elaborado por los autores

Realizar pruebas de aceptación

Tabla 6 Pruebas de aceptación

Requerimientos iniciales	
Consulta de noticias de interés	X
Acceso a servicios disponibles en la web de la universidad	X
Acceso a la galería de la universidad	X
Ingreso a las redes sociales de la universidad	X
Información general de la universidad Tecnológica Israel	X
Creación de consola de administración web	X

Fuente: Elaborado por los autores

Revisión de las pruebas de aceptación

Con las pruebas de aceptación realizadas podemos difundir conocimiento de las pruebas de aceptación y los requisitos del sistema al equipo y permitir que los miembros del equipo comenten sobre posibles mejoras en la calidad.

Día de trabajo inicial

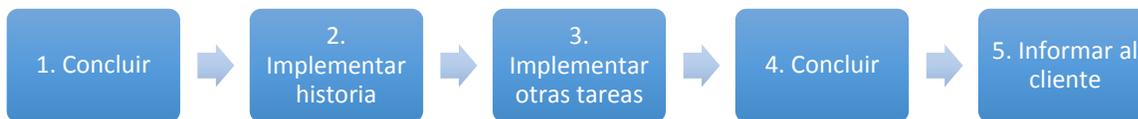


Figura 16 Día de trabajo inicial

Fuente: Elaborado por los autores

Conclusiones Es una sesión interactiva para comunicar el progreso y problemas dentro del equipo. La conclusión suele realizarse como la primera y/o la última actividad de la jornada laboral

Implementación de Historias Es una selección opcional de implementación para Trial día. Si el equipo selecciona esta opción para el ensayo, producirá mediante la aplicación de los requisitos

funcionales seleccionados en el análisis de necesidades durante el día de planificación. Así, el equipo necesita realizar actividades de implementación.

Informar al cliente El objetivo de esta tarea es proporcionar una vista del progreso al cliente, y dar al cliente una posibilidad de dar retroalimentación sobre las características implementadas y guiar el desarrollo.

Producción

Las etapas de la fase de producción se presentan en las siguientes figuras.

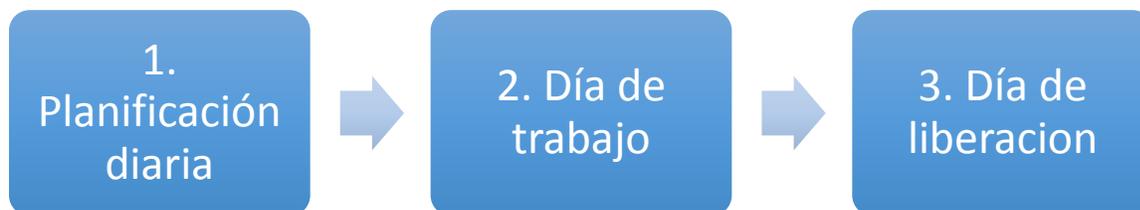


Figura 17 Etapas de producción

Fuente: Elaborado por los autores

Día de planificación



Figura 18 Día de planificación

Fuente: Elaborado por los autores

Taller de post-iteración. El propósito del Taller de Post-iteración es mejorar iterativamente el proceso de desarrollo de software para encajar mejor las necesidades del actual equipo de proyecto de software. Esto incluye la identificación fortalezas y debilidades en proceso, así como la generación de acciones de mejora para la siguiente iteración.

Análisis de requerimientos. El propósito del Análisis de Requisitos es priorizar y analizar cuidadosamente los requisitos seleccionados para cada iteración. Durante esta tarea se garantiza que los requisitos que la mayoría de los valores de negocio se identifican, y esos requisitos son correctamente entendidos.

Planificación de la iteración. El objetivo del Plan de Iteración es generar la programación y el contenido de la iteración a ejecutar. El contenido es definido en términos de tareas que son órdenes de trabajo para el equipo.

Generación de Pruebas de Aceptación. El propósito de las Pruebas de Aceptación es verificar que los requisitos que el cliente ha establecido para el software correctamente. Las pruebas de aceptación se generan posteriormente al día de la planificación, las pruebas de aceptación del día de lanzamiento se ejecutan con el cliente y los problemas encontrados (es decir, defectos y mejoras) son documentados.

Revisión de la Prueba de Aceptación. El propósito de esta tarea es difundir conocimiento de las Pruebas de Aceptación y los requisitos del sistema a todo el equipo y permitir que los miembros del equipo comenten sobre las pruebas de aceptación para mejorar su calidad.

Día de trabajo

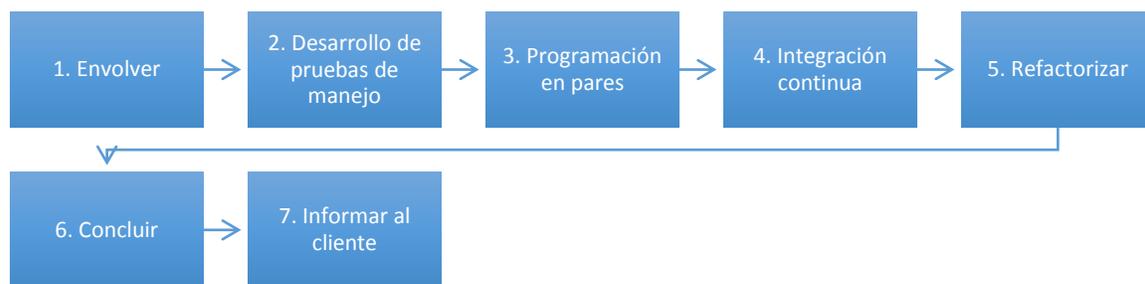


Figura 19 Día de trabajo

Fuente: Elaborado por los autores

Envolver. Es una sesión interactiva para comunicar el progreso y problemas dentro del equipo. La conclusión suele realizarse como la primera y / o la última actividad de la jornada de trabajo (resumen de la mañana y Tarde de recapitulación).

Desarrollo Impulsado por Pruebas. En TDD las pruebas unitarias se escriben antes del código de programa. El código de programa se desarrolla para trabajar con las pruebas ya escritas. El propósito TDD es dar a los desarrolladores confianza que el código que producen las obras y guían el diseño del código para clarificar estructura más fácilmente.

La programación de pares es un estilo de programación donde dos desarrolladores codifican juntos. El conductor es la persona que usa el teclado y el ratón durante una sesión de programación de pares, mientras que el navegador se sienta delante del computador al lado del conductor. El propósito de la programación de pares es mejorar comunicación, difundir el conocimiento dentro del equipo y asegurar la calidad del código.

Integración continua. El propósito de la integración continua es integrar continuamente el nuevo código con el código existente en un código repositorio. Al integrar continuamente las integraciones masivas se pueden evitar que de lo contrario tomar mucho tiempo y esfuerzo.

La refactorización es el proceso en el cual se mejora la capacidad interna de la estructura sin modificar su comportamiento externo. Con pequeñas mejoras en el código, la refactorización asegura que el software sea más modificable, extensible y legible.

Informar al Cliente. El objetivo de esta tarea es proporcionar una vista del progreso al cliente y obtener la retroalimentación necesaria sobre las características implementadas.

Día de lanzamiento

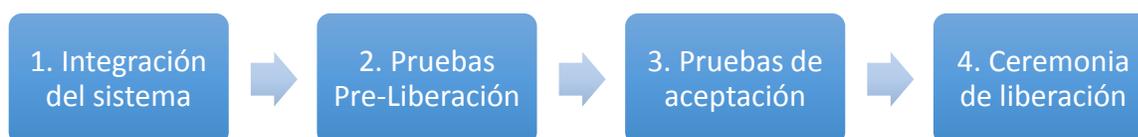


Figura 20 Día de liberación

Fuente: Elaborado por los autores

Integración de sistema. En el caso del proyecto multi – equipo, el propósito de esta tarea consiste en integrar subsistemas, generados en equipos separados, en un solo producto.

Pruebas de pre-liberación. El propósito de esta tarea es asegurarse que el software que se produce esté listo para la prueba de aceptación y la respectiva liberación.

Las Pruebas Previas a la Liberación se llevan a cabo dentro del equipo ejecutando todas las pruebas de aceptación.

Prueba de aceptación. El propósito de las pruebas de aceptación es verificar que los requisitos que el cliente ha establecido para el software, se implementen correctamente. Durante el día de lanzamiento, los ensayos de aceptación se ejecutan y se registran (es decir, defectos y mejoras) quedan documentados.

Las ceremonias de lanzamiento son las tareas finales antes de hacer un lanzamiento de software. En la práctica, las ceremonias de liberación consisten en dos actividades esenciales; La auditoría de lanzamiento y la creación de línea de base. Estas actividades se realizan para confirmar que todo ha sido hecho correctamente en la iteración actual y la base para garantiza un mayor desarrollo.

Servidor REST (Iteración 1)

Día de planificación

Taller de post-iteración. En esta iteración no es necesaria el taller post iteración ya que es la primera interacción y no existe ninguna previa para realizar la integración.

Análisis de requerimientos.

Seleccionamos los requerimientos a ser implementados

- Ingreso a la consola de administración

- Configuración de noticias

- Configuración de servicios

- Configuración de galería

- Configuración de redes sociales

- Configuración de información

Discusión y análisis de los requerimientos seleccionados uno a la vez, soportados con la ilustración de la interfaz de usuario como material de soporte

Ingreso a la consola de administración

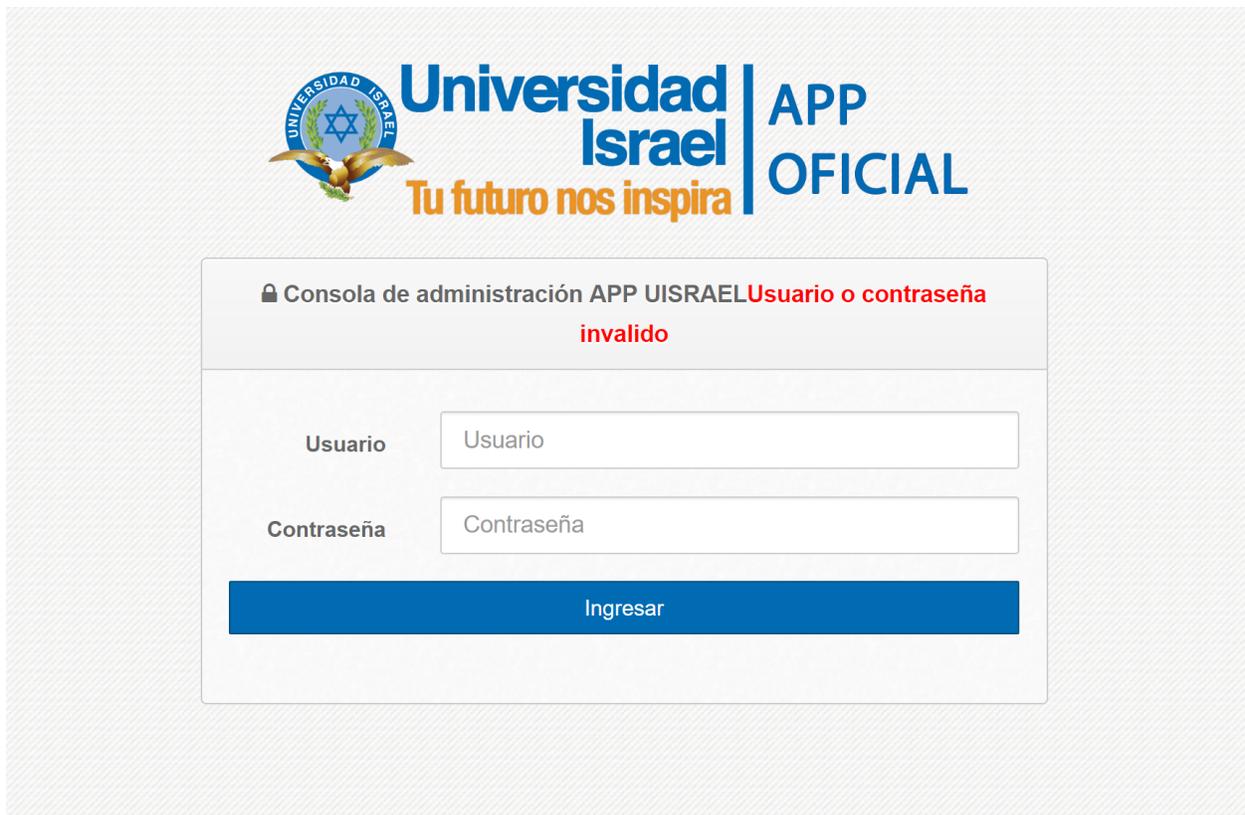


Figura 21 Ingreso a consola

Fuente: Elaborado por los autores

Menú de secciones

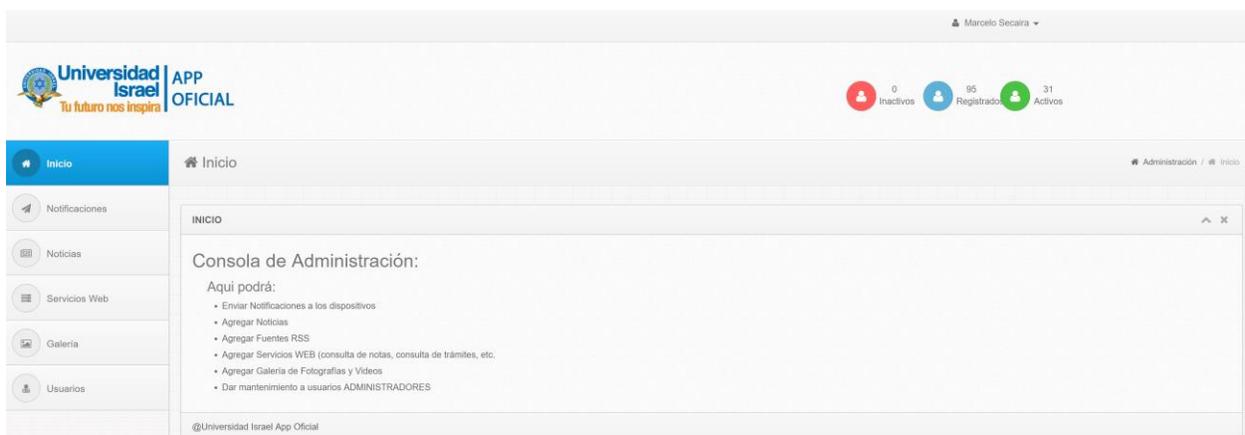


Figura 22 Menú principal

Fuente: Elaborado por los autores

Envío de notificaciones

ENVIAR NOTIFICACIÓN

Título

Contenido

Figura 23 Envío de notificaciones

Fuente: Elaborado por los autores

Configuración de noticias

ADMINISTRACIÓN NOTICIAS

Fuentes RSS

Agregar Fuente RSS

Nombre

URL

Id	Nombre	URL	Acción
1	UNIVERSIDAD ISRAEL	http://uisrael.edu.ec/feed/	<input type="button" value="Test"/>
2	EL UNIVERSO - ECUADOR	http://m.eluniverso.com/rss/ecuador.xml	<input type="button" value="Eliminar"/> <input type="button" value="Test"/>

Figura 24 Configuración noticias

Fuente: Elaborado por los autores

Configuración de servicios

ADMINISTRACIÓN SERVICIOS WEB

Listado de Servicios

ID	Activo	Nombre	Descripción	URL	Icono	Creado	Editar
6	TRUE	VOTACIÓN PROCESO ELE	Módulo de voto electrónico	http://190.11.245.242:9090/SIGE/MVE/Default.aspx		2017-08-17 10:16:09	<input type="button" value="Editar"/>
5	TRUE	Evaluación Docentes	Servicio en el cual, los estudiantes podrán evaluar a los docentes.	http://190.11.245.242:9090/EvaluacionDocentes/Login.aspx		2017-08-09 19:17:39	<input type="button" value="Editar"/>
4	TRUE	CARGA ACADEMICA	Los estudiantes podrán acceder a la consulta de la carga académica.	http://190.11.245.242:9090/CargaAcademica/		2017-08-09 19:05:55	<input type="button" value="Editar"/>
1	TRUE	CALIFICACIONES	Servicio para la consulta de las calificaciones de los estudiantes de la Uisrael	http://190.11.245.242:9090/WebConsultaNotasUisrael/consulta.aspx		2016-11-27 22:14:49	<input type="button" value="Editar"/>

Figura 25 Configuración servicios

Fuente: Elaborado por los autores

Configuración de galería

ADMINISTRACIÓN GALERÍA			
Lista de fotografías en galería			
Agregar Item			
Imagen ID	Descripción	Fotografía	Editar
16	Jornadas tecnológicas 4to grupo		Editar
15	Jornadas tecnológicas 3er grupo		Editar
14	Jornadas tecnológicas 2do grupo		Editar
13	Jornadas tecnológicas Primer Grupo		Editar

Figura 26 Configuración Galería

Fuente: Elaborado por los autores

Planificación de la iteración.

La planificación de esta iteración se derivan las siguientes tareas:

Determinar la velocidad del equipo

Refinar la tarea con el uso de gráficos

Ingreso a la consola de administración

Tabla 7 Ingreso a consola de administración

Tarjeta de tarea: Ingreso a la consola de administración							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
CW-01	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción:							
El ingreso a la consola de administración permite la digitación del usuario y contraseña para validar el acceso e ingresar a la plataforma.							

Fecha:	Estado:	Comentarios:
01/11/2016	Definido	Inicio del desarrollo
02/11/2016	Implementado	Desarrollo Primera versión
03/11/2016	Realizado	Primera ejecución
05/11/2016	Verificado	Verificación de funcionalidad
05/11/2016	Pospuesto/ cancelado	

Fuente: Elaborado por los autores

Configuración de noticias

Tabla 8 Configuración de noticias

Tarjeta de tarea: Ingreso de noticias							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
CW-02	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción:							
Esta ventana de la consola web permite crear noticias de interés para la universidad. Las cuales estarán disponibles en el servidor REST							
Fecha:	Estado:	Comentarios:					
06/11/2016	Definido	Inicio del desarrollo					
07/11/2016	Implementado	Desarrollo Primera versión					
09/11/2016	Realizado	Primera ejecución					
10/11/2016	Verificado	Verificación de funcionalidad					
	Pospuesto/ cancelado						

Fuente: Elaborado por los autores

Configuración de servicios

Tabla 9 Configuración de servicios

Tarjeta de tarea: Ingreso de Servicios							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
CW-03	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción:							
<p>Esta página permite ingresar y configurar los servicios del portal web de la Universidad Tecnológica Israel:</p> <ul style="list-style-type: none"> • Consulta de calificaciones • Inscripciones • Balcón de servicios 							
Fecha:	Estado:	Comentarios:					
11/11/2016	Definido	Inicio del desarrollo					
12/11/2016	Implementado	Desarrollo Primera versión					
14/11/2016	Realizado	Primera ejecución					
15/11/2016	Verificado	Verificación de funcionalidad					
	Pospuesto/ cancelado						

Fuente: Elaborado por los autores

Configuración de galería

Tabla 10 Configuración de galería

Tarjeta de tarea: Configuración de Galería							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
CW-04	<u>Nuevo</u>	Fácil	Fácil				
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				

	Mejora	Difícil	Difícil				
Descripción:							
La página de galería nos permite ingresar fotografías, imágenes y videos que estarán disponibles en el servidor web							
Fecha:	Estado:		Comentarios:				
16/11/2016	Definido		Inicio del desarrollo				
17/11/2016	Implementado		Desarrollo Primera versión				
19/11/2016	Realizado		Primera ejecución				
20/11/2016	Verificado		Verificación de funcionalidad				
	Pospuesto/ cancelado						

Fuente: Elaborado por los autores

Publicación del sitio web

Tabla 11 Publicación del sitio web

Tarjeta de tarea: Publicación del sitio web							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
CW-05	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción:							
<p>En la tarea publicamos la consola de administración web con sus respectivos servicios. Para lo que seguimos los siguientes pasos.</p> <ul style="list-style-type: none"> • Adquisición de un dominio • Adquisición del hosting • Creación de la base de datos • Publicación de la consola web 							

Fecha:	Estado:	Comentarios:
20/11/2016	Listo	Dominio
21/11/2016	Listo	Hosting
22/11/2016	Listo	Base de datos
23/11/2016	Listo	Publicación de sitio
24/11/2016	Listo	Implementación servicios web
	Pospuesto/ cancelado	

Fuente: Elaborado por los autores

Generación de Pruebas de Aceptación.

Las pruebas de aceptación se las establece en base a los requisitos del cliente los detallamos a continuación:

Tabla 12 Pruebas ingreso a la consola de administración

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: CW-01
Tarea: Ingreso a la consola de administración
Fecha de realización: 01 de octubre 2016
Fecha de Ejecución: 01 de octubre 2016
Descripción (que funcionalidad debe implementar para ser aceptada) <ol style="list-style-type: none"> 1. El menú de inicio de sesión debe mostrarse como en la documentación. 2. La página debe mostrar una caja de texto para el usuario 3. La página debe mostrar una caja de texto para la contraseña 4. La página debe mostrar un botón para ingresar a la consola 5. Funcionalidad botón inicia. Envía usuario y contraseña <ol style="list-style-type: none"> 5.1. Usuario y contraseña correctos 5.2. Usuario o contraseña incorrectos
Resultados esperados (Que esperamos como resultado) <ol style="list-style-type: none"> 1. Inspección visual 2. Inspección visual 3. Inspección visual 4. Verificación de usuario y contraseña <ol style="list-style-type: none"> 4.1. Ingreso correcto “bienvenido” 4.2. usuario o contraseña no validos “intente nuevamente”

Fuente: Elaborado por los autores

Tabla 13 Pruebas ingresos de noticias

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: CW-02
Tarea: Ingreso de noticias
Fecha de realización: 05 octubre 2016
Fecha de Ejecución: 05 de octubre 2016
Descripción (que funcionalidad debe implementar para ser aceptada) <ol style="list-style-type: none"> 1. El menú de inicio de sesión debe mostrarse como en la documentación. 2. La página debe mostrar una caja de esto para poner el título de la noticia 3. La página debe mostrar una caja de texto para ingresar la URL del RSS 4. La página debe tener un botón para grabar la fuente
Resultados esperados (Que esperamos como resultado) <ol style="list-style-type: none"> 1. Inspección visual 2. Inspección visual 3. Inspección visual 4. La aplicación ingresa el servicio a la base de datos de las fuentes de noticias

Fuente: *Elaborado por los autores*

Tabla 14 Pruebas configuración de servicios

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: CW-03
Tarea: Configuración de servicios
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016
Descripción (que funcionalidad debe implementar para ser aceptada) <ol style="list-style-type: none"> 1. La página debe mostrarse como en la documentación. 2. La página debe mostrar una caja de esto para poner el título del servicio 3. La página debe mostrar una caja de texto para ingresar la URL del servicio 4. La página debe tener un botón para grabar la fuente
Resultados esperados (Que esperamos como resultado) <ol style="list-style-type: none"> 1. Inspección visual 2. Inspección visual

<ol style="list-style-type: none"> 3. Inspección visual 4. La aplicación ingresa el servicio a la base de datos con los servicios disponibles en el portal web.

Fuente: Elaborado por los autores

Tabla 15 Pruebas configuración de galería

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: CW-04
Tarea: Configuración de Galería
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016
Descripción (que funcionalidad debe implementar para ser aceptada) <ol style="list-style-type: none"> 1. La página debe mostrarse como en la documentación. 2. La página debe mostrar una caja de esto para poner el título de la galería 3. La página debe mostrar un contenedor para subir el archivo 4. La página debe tener un botón para grabar la fuente
Resultados esperados (Que esperamos como resultado) <ol style="list-style-type: none"> 1. Inspección visual 2. Inspección visual 3. Inspección visual 4. La aplicación ingresa el servicio a la base de datos con las imágenes disponibles.

Fuente: Elaborado por los autores

Revisión de la Prueba de Aceptación.

Con los conocimientos adquiridos en base a los requisitos del cliente, las pruebas de aceptación y las tareas definidas, difundimos este conocimiento a todos los miembros del equipo para que puedan sugerir posibles mejoras.

Día de trabajo

Envolver. Es una sesión interactiva para comunicar el progreso y problemas dentro del equipo. La conclusión suele realizarse como la primera y/o la última actividad de la jornada de trabajo (resumen de la mañana y tarde de recapitulación).

Desarrollo Impulsado por Pruebas. En TDD las pruebas unitarias se escriben antes del código de programa. El código de programa se desarrolla para trabajar con las pruebas ya escritas. El propósito TDD es dar a los desarrolladores confianza que el código que producen las obras y guían el diseño del código para clarificar estructura más fácilmente.

Integración continua. El propósito de la integración continua es integrar continuamente el nuevo código con el código existente en un código repositorio. Al integrar continuamente las integraciones masivas se pueden evitar que de lo contrario tomar mucho tiempo y esfuerzo.

La refactorización es el proceso de mejorar la capacidad interna de la estructura sin modificar su comportamiento externo. Con pequeñas mejoras en el código, la refactorización asegura que el software sea más modificable, extensible y legible.

Informar al Cliente. El objetivo de esta tarea es proporcionar una vista del progreso al cliente, y dar al cliente una posibilidad de dar retroalimentación sobre las características implementadas y guiar el desarrollo.

Día de liberación

Integración del sistema

En la presente iteración es la primera en ser desarrollada por lo que no es necesario la integración con el sistema o proyecto final. Más bien se realizan pruebas de funcionamiento unitario de esta iteración.

Resolución de problemas

A continuación de las pruebas unitarias determinamos si existen problemas o fallas que necesitan ser resueltas.

Reunión de entrega

Tabla 16 Verificación de auditorías

Lista de Verificación de auditorías			
fecha:	31 de enero del 2017	liberación #	
Requerimientos	SI	No	NA
1. Todas las historias tienen el estado de "verificado"	X		
2. Todas las tareas tienen el estado de "verificado"	X		
3. Se han realizado pruebas previas a la liberación (y se han documentado problemas)	X		
4. Se han realizado pruebas de liberación (y se han documentado problemas)	X		
5. Los documentos están al día (revisión)	X		
6. Las métricas están actualizadas (bug-list)	X		
7. El repositorio de control de versiones está actualizado (se han registrado todos los cambios relevantes)	X		
Comentarios: Todas las auditorías han sido revisadas y documentadas de acuerdo a la metodología establecida			
seleccione uno: <input type="checkbox"/> Los resultados revisados satisfacen los requisitos y son aceptados. <input type="checkbox"/> Los resultados revisados son aceptados con cambios (Ver listas de resumen de deficiencia, página 2). <input type="checkbox"/> Los resultados revisados no cumplen con los requisitos (ver listas de resumen de deficiencia, página 2)			
Aprobado por: _____ Fecha: 31 de enero del 2017			

Proyecto: lista de control de liberación de auditorías fecha: 30/09/2016

Tabla 17 Listado de deficiencias

lista de resumen de deficiencias		
Requerimiento #	Descripción	Acción de corrección

Aplicación móvil Android (Iteración 2)

Día de planificación

Taller de post-iteración. En esta iteración no es necesaria el taller post iteración ya que es la primera interacción y no existe una iteración previa para realizar la integración.

Análisis de requerimientos.

Seleccionamos los requerimientos a ser implementados

- Visualización de noticias
- Visualización de servicios
- Visualización de galería
- Visualización de redes sociales
- Configuración de información

Discusión y análisis de los requerimientos seleccionados uno a la vez, soportados con la ilustración de la interfaz de usuario como material de soporte

- Visualización de noticias

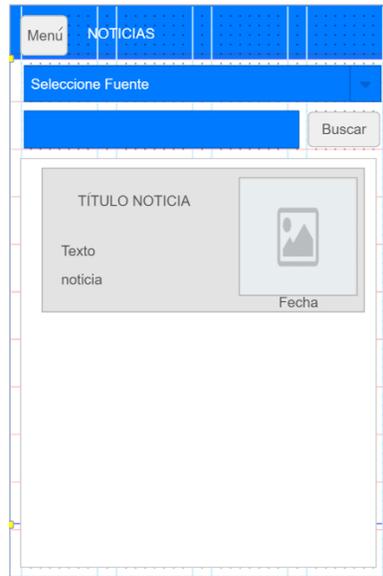


Figura 27 Visualización de noticias

Fuente: Elaborado por los autores

Visualización de servicios

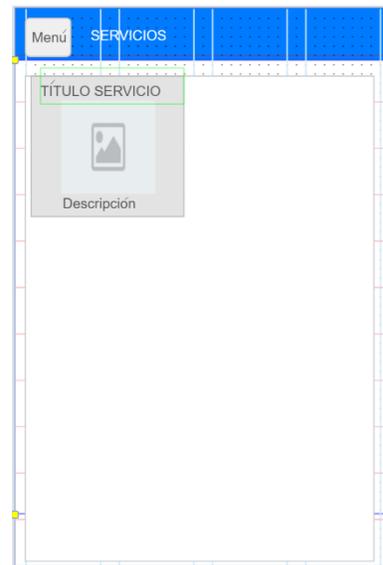


Figura 28 Visualización de servicios

Fuente: Elaborado por los autores

Visualización de galería

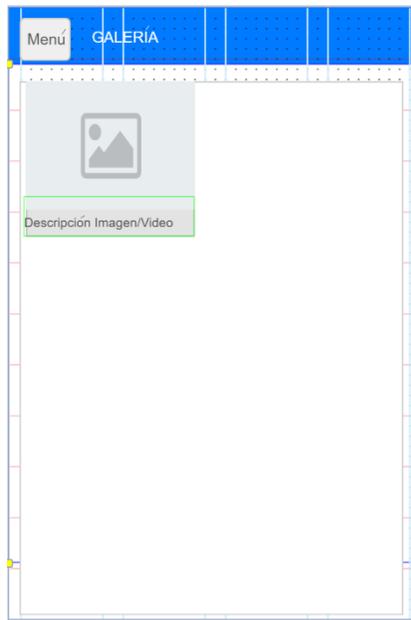


Figura 29 Visualización de galería

Fuente: Elaborado por los autores

Visualización de redes sociales

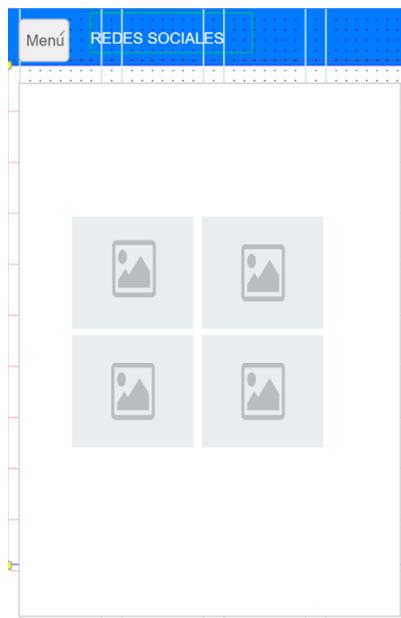


Figura 30 Visualización de redes sociales

Fuente: Elaborado por los autores

Visualización de información



Figura 31 Visualización de información

Fuente: Elaborado por los autores

Planificación de la iteración.

La planificación de esta iteración se derivan las siguientes tareas:

Determinar la velocidad del equipo

Refinar la tarea con el uso de gráficos

Ingreso a la vista de inicio

Tabla 18 Vista de inicio

Tarjeta de tarea: Vista de inicio							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
AD-01	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción: Esta es la vista inicial de nuestra aplicación							
<ul style="list-style-type: none"> Es la vista principal de nuestra aplicación y es la encargada de guiarnos por todas las vistas de la aplicación 							

Fecha:	Estado:	Comentarios:
01/10/2016	Definido	Inicio del desarrollo
03//10/2016	Implementado	Desarrollo Primera versión
04/10/2016	Realizado	Primera ejecución
05/10/2016	Verificado	Verificación de funcionalidad
	Pospuesto/ cancelado	

Fuente: Elaborado por los autores

Ingreso vista de noticias

Tabla 19 Vista de noticias

Tarjeta de tarea: Vista de Noticias							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
AD-02	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción: esta vista despliega un listado de noticias							
<ul style="list-style-type: none"> • La vista de noticias muestra las noticias publicadas en el portal web de la Universidad • La vista de noticias muestra las noticias de otras fuentes de interés (Periódicos locales) • La vista de noticias no permite seleccionar la noticia para ampliar la imagen y el desplegar el contenido más detallado 							
Fecha:	Estado:	Comentarios:					
06/10/2016	Definido	Inicio de la programación					
07/10/2016	Implementado	Desarrollo Primera versión					
09/10/2016	Realizado	Primera ejecución					
10/10/2016	Verificado	Verificación de funcionalidad					
	Pospuesto/ cancelado						

Fuente: Elaborado por los autores

Ingreso vista de servicios

Tabla 20 Vista de servicios

Tarjeta de tarea: Vista de servicios

Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
AD-03	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción: esta vista despliega un listado de servicios disponibles en el portal web							
<ul style="list-style-type: none"> • La vista de servicios muestra los servicios disponibles en el portal web de la Universidad • La vista de servicios permite seleccionar el tipo de servicios y nos dirige al servicio solicitado <ul style="list-style-type: none"> ○ Consulta de calificaciones ○ Inscripciones ○ Balcón de servicios 							
Fecha:	Estado:	Comentarios:					
11/10/2016	Definido	Inicio del desarrollo					
12/10/2016	Implementado	Desarrollo Primera versión					
13/10/2016	Realizado	Primera ejecución					
15/10/2016	Verificado	Verificación de funcionalidad					
	Pospuesto/ cancelado						

Fuente: Elaborado por los autores

Ingreso vista de galería

Tabla 21 Vista de galería

Tarjeta de tarea: Vista de galería							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
AD-04	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción: esta vista despliega la galería							
<ul style="list-style-type: none"> • Inicia desplegando una colección de imágenes y videos de la universidad • Al seleccionar la imagen podemos amplia la imagen 							

<ul style="list-style-type: none"> En caso de video podemos reproducir el mismo 		
Fecha:	Estado:	Comentarios:
16/10/2016	Definido	Inicio del desarrollo
17//10/2016	Implementado	Desarrollo Primera versión
19/10/2016	Realizado	Primera ejecución
20/10/2016	Verificado	Verificación de funcionalidad
	Pospuesto/ cancelado	

Fuente: Elaborado por los autores

Ingreso vista de redes sociales

Tabla 22 Vista redes sociales

Tarjeta de tarea: Vista de redes sociales							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
AD-05	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción: la vista de redes sociales muestra listado de accesos a cada uno de las redes							
<ul style="list-style-type: none"> Esta vista presenta una colección de accesos a las redes sociales a las que la universidad se encuentra suscrita. <ul style="list-style-type: none"> Facebook Twitter Instagram YouTube 							
Fecha:	Estado:	Comentarios:					
21/10/2016	Definido	Inicio del desarrollo					
22//10/2016	Implementado	Desarrollo Primera versión					
24/10/2016	Realizado	Primera ejecución					
25/10/2016	Verificado	Verificación de funcionalidad					
	Pospuesto/ cancelado						

Fuente: Elaborado por los autores

Ingreso vista de información

Tabla 23 Vista de información

Tarjeta de tarea: vista de información							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
AD-06	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción:							
<ul style="list-style-type: none"> • La vista muestra la localización geográfica por medio de un mapa. • La vista tiene disponible información de los contactos de la universidad • La vista tiene la historia de la universidad Tecnológica Israel 							
Fecha:	Estado:		Comentarios:				
26/10/2016	Definido		Inicio del desarrollo				
27//10/2016	Implementado		Desarrollo Primera versión				
29/10/2016	Realizado		Primera ejecución				
30/10/2016	Verificado		Verificación de funcionalidad				
	Pospuesto/ cancelado						

Fuente: Elaborado por los autores

Generación de Pruebas de Aceptación.

Las pruebas de aceptación se las establece en base a los requisitos del cliente los detallamos a continuación:

Tabla 24 Pruebas vista de noticias

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: AD-01
Tarea: Vista de noticias
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016
Descripción (que funcionalidad debe implementar para ser aceptada)
<ol style="list-style-type: none"> 1. La vista de noticias debe mostrarse como en la documentación 2. La vista tiene un combo box con los tipos de fuentes de información <ol style="list-style-type: none"> 2.1. Función seleccionar fuente universidad

<ul style="list-style-type: none"> 2.2. Función seleccionar otra fuente 3. La vista tiene una caja de texto 3.1. Función filtrar las noticias por el texto ingresado 4. La vista tiene una tabla con todas las noticias 4.1. Selección de noticia de la tabla
<p>Resultados esperados (Que esperamos como resultado)</p> <ul style="list-style-type: none"> 1. Inspección visual 2. Inspección visual 2.1. Despliega noticias fuente de la universidad 2.2. Despliega noticias otra fuente 3. Inspección visual 3.1. Muestra las noticias que concuerdan con el texto ingresado 4. Inspección visual 4.1. Abre una nueva vista con la imagen ampliada y la información completa

Fuente: Elaborado por los autores

Tabla 25 Pruebas vista de servicios

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: AD-02
Tarea: Vista de servicios
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016
<p>Descripción (que funcionalidad debe implementar para ser aceptada)</p> <ul style="list-style-type: none"> 1. La vista de servicios debe mostrarse como en la documentación 2. La vista tiene una colección de iconos uno por cada servicio 2.1. Función seleccionar icono servicio
<p>Resultados esperados (Que esperamos como resultado)</p> <ul style="list-style-type: none"> 1. Inspección visual 2. Inspección visual 2.1. Despliega servicios disponibles en el portal web de la universidad

Fuente: Elaborado por los autores

Tabla 26 Pruebas vista de galería

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: AD-03
Tarea: Vista de Galería
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016
Descripción (que funcionalidad debe implementar para ser aceptada) <ol style="list-style-type: none">1. La vista de servicios debe mostrarse como en la documentación2. La vista tiene una colección de iconos uno por cada imagen o video<ol style="list-style-type: none">2.1. Función seleccionar icono servicio
Resultados esperados (Que esperamos como resultado) <ol style="list-style-type: none">1. Inspección visual2. Inspección visual<ol style="list-style-type: none">2.1. Reproduce el video o amplía la imagen

Fuente: Elaborado por los autores

Tabla 27 Pruebas vista redes sociales

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: AD-04
Tarea: Vista de Redes sociales
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016
Descripción (que funcionalidad debe implementar para ser aceptada) <ol style="list-style-type: none">1. La vista de redes sociales debe mostrarse como en la documentación2. La vista tiene una colección de iconos uno por cada red social<ol style="list-style-type: none">2.1. Función seleccionar icono de red social
Resultados esperados (Que esperamos como resultado) <ol style="list-style-type: none">1. Inspección visual2. Inspección visual

2.1. Despliega vista con un navegador que tiene direccionado al sitio web de la red social

Fuente: Elaborado por los autores

Tabla 28 Pruebas vista de información

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: AD-05
Tarea: Vista Información
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016
Descripción (que funcionalidad debe implementar para ser aceptada) <ol style="list-style-type: none">1. La vista de información debe mostrarse como en la documentación2. La vista debe contener un mapa utilizando el API de GOOGLE3. Etiquetas de texto con información dirección4. Etiqueta de texto con información de números de teléfono5. Etiqueta de texto que muestra la información de la historia de la universidad
Resultados esperados (Que esperamos como resultado) <ol style="list-style-type: none">1. Inspección visual2. Inspección visual3. Inspección visual4. Inspección visual5. Inspección visual

Fuente: Elaborado por los autores

Revisión de la Prueba de Aceptación.

Con los conocimientos adquiridos en base a los requisitos del cliente, las pruebas de aceptación y las tareas definidas difundimos este conocimiento a todos los miembros del equipo para que puedan sugerir posibles mejoras.

Día de trabajo

Envolver. Es una sesión interactiva para comunicar el progreso y problemas dentro del equipo. La conclusión suele realizarse como la primera y/o la última actividad de la jornada de trabajo (resumen de la mañana y tarde de recapitulación).

Desarrollo Impulsado por Pruebas. En TDD las pruebas unitarias se escriben antes del código de programa. El código de programa se desarrolla para trabajar con las pruebas ya escritas. El propósito TDD es dar a los desarrolladores confianza para que codifiquen según los requerimientos solicitados.

Integración continua. El propósito de la integración continua es integrar continuamente el nuevo código con el código existente en un código repositorio. Al integrar continuamente las integraciones masivas se pueden evitar que de lo contrario tomar mucho tiempo y esfuerzo.

La refactorización es el proceso de mejorar la capacidad interna de la estructura sin modificar su comportamiento externo. Con pequeñas mejoras en el código, la refactorización asegura que el software sea más modificable, extensible y legible.

Informar al Cliente. El objetivo de esta tarea es proporcionar una vista del progreso al cliente, y dar al cliente una posibilidad de dar retroalimentación sobre las características implementadas y guiar el desarrollo.

Día de liberación

Integración del sistema

En la presente iteración es la primera en ser desarrollada por lo que no es necesario la integración con el sistema o proyecto final. Más bien se realizan pruebas de funcionamiento unitario de esta iteración.

Resolución de problemas

A continuación de las pruebas unitarias determinamos si existen problemas o fallas que necesitan ser resueltas.

Reunión de entrega

Tabla 29 Verificación de auditorías

Lista de Verificación de auditorías			
fecha:	liberación #		
Requerimientos	SI	No	NA
1. todas las historias tienen el estado de "verificado"	X		
2. Todas las tareas tienen el estado de "verificado"	X		
3. Se han realizado pruebas previas a la liberación (y se han documentado problemas)	X		
4. Se han realizado pruebas de liberación (y se han documentado problemas)	X		
5. Los documentos están al día (revisión)	X		
6. Las métricas están actualizadas (bug-list, LOCs ...)	X		
7. El repositorio de control de versiones está actualizado (se han registrado todos los cambios relevantes)	X		
Comentarios:			
La iteración de la aplicación móvil con sistema operativo Android cumple con todas las auditorías			
seleccione uno:			
<input type="checkbox"/> Los resultados revisados satisfacen los requisitos y son aceptados.			
<input type="checkbox"/> Los resultados revisados son aceptados con cambios (Ver listas de resumen de deficiencia, página 2).			
<input type="checkbox"/> Los resultados revisados no cumplen con los requisitos (ver listas de resumen de deficiencia, página 2)			
Aprobado por: _____ Fecha: _____			

Fuente: Elaborado por los autores

Proyecto:

Lista de control de liberación de auditorías

fecha: 30/10/2016

Tabla 30 lista de resumen de deficiencias

lista de resumen de deficiencias		
Requerimiento #	Descripción	Acción de corrección

Fuente: Elaborado por los autores

Aplicación móvil IOS (Iteración 3)

Día de planificación

Taller de post-iteración. En esta iteración no es necesaria el taller post iteración ya que es la primera interacción y no existe una iteración previa para realizar la integración.

Análisis de requerimientos.

Seleccionamos los requerimientos a ser implementados

- Visualización inicial
- Visualización de noticias
- Visualización de servicios
- Visualización de galería
- Visualización de redes sociales
- Visualización de información

Discusión y análisis de los requerimientos seleccionados uno a la vez, soportados con la Ilustración de la interfaz de usuario como material de soporte

Visualización de inicio

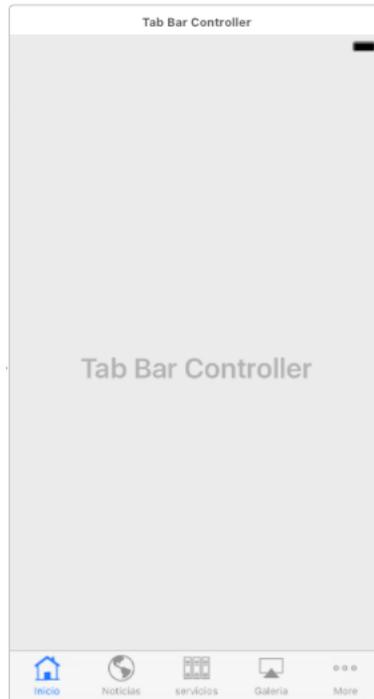


Figura 32 Vista inicial IOS

Fuente: Elaborado por los autores

Visualización de noticias



Figura 33 Vista de Noticias IOS

Fuente: Elaborado por los autores

Visualización de servicios

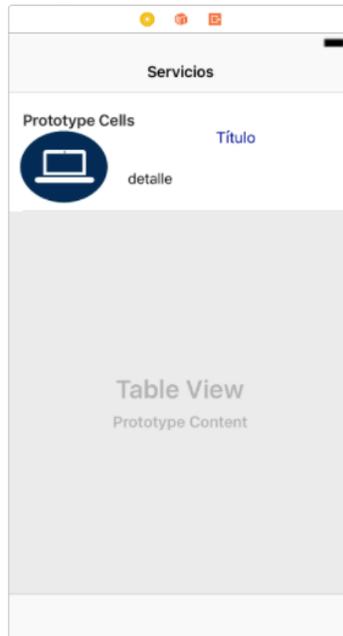


Figura 34 Visualización de servicios IOS

Fuente: Elaborado por los autores

Visualización de galería



Figura 35 Visualización Galería IOS

Fuente: Elaborado por los autores

Visualización de redes sociales



Figura 36 Redes Sociales IOS

Fuente: Elaborado por los autores

Visualización de información



Figura 37 Información IOS

Fuente: Elaborado por los autores

Planificación de la iteración.

La planificación de esta iteración se derivan las siguientes tareas:

Determinar la velocidad del equipo

- Se establece para cada miembro del equipo el trabajo se debe emplear 4 horas diarias

Refinar la tarea con el uso de gráficos

Ingreso a la vista de inicio

Tabla 31 Vista de inicio

Tarjeta de tarea: Vista de inicio							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
IOS-01	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción: Esta es la vista inicial de nuestra aplicación							
<ul style="list-style-type: none">• Es la vista principal que servirá como menú de navegación.							
Fecha:	Estado:	Comentarios:					
01/10/2016	Definido	Inicio del desarrollo					
03//10/2016	Implementado	Desarrollo Primera versión					
04/10/2016	Realizado	Primera ejecución					
05/10/2016	Verificado	Verificación de funcionalidad					
	Pospuesto/ cancelado						

Fuente: Elaborado por los autores

Ingreso vista de noticias

Tabla 32 Vista de noticias

Tarjeta de tarea: Vista de Noticias							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		

IOS-02	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción: esta vista despliega un listado de noticias							
<ul style="list-style-type: none"> • La vista de noticias muestra las noticias publicadas en el portal web de la Universidad • La vista de noticias muestra las noticias de otras fuentes de interés (Periódicos locales) • La vista de noticias no permite seleccionar la noticia para ampliar la imagen y el desplegar el contenido más detallado 							
Fecha:	Estado:		Comentarios:				
06/10/2016	Definido		Inicio de la programación				
07/10/2016	Implementado		Desarrollo Primera versión				
09/10/2016	Realizado		Primera ejecución				
10/10/2016	Verificado		Verificación de funcionalidad				
	Pospuesto/ cancelado						

Fuente: Elaborado por los autores

Ingreso vista de servicios

Tabla 33 Vista de servicios

Tarjeta de tarea: Vista de servicios							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
IOS-03	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción: esta vista despliega un listado de servicios disponibles en el portal web							
<ul style="list-style-type: none"> • La vista de servicios muestra los servicios disponibles en el portal web de la Universidad • La vista de servicios permite seleccionar el tipo de servicios y nos dirige al servicio solicitado <ul style="list-style-type: none"> ○ Consulta de calificaciones ○ Inscripciones ○ Balcón de servicios 							
Fecha:	Estado:		Comentarios:				

11/10/2016	Definido	Inicio del desarrollo
12/10/2016	Implementado	Desarrollo Primera versión
13/10/2016	Realizado	Primera ejecución
15/10/2016	Verificado	Verificación de funcionalidad
	Pospuesto/ cancelado	

Fuente: Elaborado por los autores

Ingreso vista de galería

Tabla 34 Vista de galería

Tarjeta de tarea: Vista de galería							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
IOS-04	<u>Nuevo</u>	Fácil	Fácil				
	Arreglo	<u>Moderado</u>	<u>Moderado</u>	5 días	5 días		
	Mejora	Difícil	Difícil				
Descripción: esta vista despliega la galería							
<ul style="list-style-type: none"> • Inicia desplegando una colección de imágenes y videos de la universidad • Al seleccionar la imagen podemos amplia la imagen • En caso de video podemos reproducir el mismo 							
Fecha:	Estado:	Comentarios:					
16/10/2016	Definido	Inicio del desarrollo					
17//10/2016	Implementado	Desarrollo Primera versión					
19/10/2016	Realizado	Primera ejecución					
20/10/2016	Verificado	Verificación de funcionalidad					
	Pospuesto/ cancelado						

Fuente: Elaborado por los autores

Ingreso vista de redes sociales

Tabla 35 Vista redes sociales

Tarjeta de tarea: Vista de redes sociales							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		

IOS-05	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción: la vista de redes sociales muestra listado de accesos a cada uno de las redes							
<ul style="list-style-type: none"> • Esta vista presenta una colección de accesos a las redes sociales a las que la universidad se encuentra suscrita. <ul style="list-style-type: none"> ○ Facebook ○ Twitter ○ Instagram ○ YouTube 							
Fecha:	Estado:		Comentarios:				
21/10/2016	Definido		Inicio del desarrollo				
22//10/2016	Implementado		Desarrollo Primera versión				
24/10/2016	Realizado		Primera ejecución				
25/10/2016	Verificado		Verificación de funcionalidad				
	Pospuesto/ cancelado						

Fuente: Elaborado por los autores

Vista de información

Tabla 36 Vista de información

Tarjeta de tarea: vista de información							
Numero / ID	Tipo	Dificultad		Esfuerzo		Prioridad	Notas
		Antes	Después	Estimado	Utilizado		
IOS-06	<u>Nuevo</u>	Fácil	Fácil	5 días	5 días		
	Arreglo	<u>Moderado</u>	<u>Moderado</u>				
	Mejora	Difícil	Difícil				
Descripción:							
<ul style="list-style-type: none"> • La vista muestra la localización geográfica por medio de un mapa. • La vista tiene disponible información de los contactos de la universidad • La vista tiene la historia de la universidad Tecnológica Israel 							
Fecha:	Estado:		Comentarios:				
26/10/2016	Definido		Inicio del desarrollo				

27//10/2016	Implementado	Desarrollo Primera versión
29/10/2016	Realizado	Primera ejecución
30/10/2016	Verificado	Verificación de funcionalidad
	Pospuesto/ cancelado	

Fuente: Elaborado por los autores

Generación de Pruebas de Aceptación.

Las pruebas de aceptación se las establece en base a los requisitos del cliente los detallamos a continuación:

Tabla 37 Pruebas de aceptación

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: IOS-01
Tarea: Vista de noticias
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016
<p>Descripción (que funcionalidad debe implementar para ser aceptada)</p> <ol style="list-style-type: none"> 1. La vista de noticias debe mostrarse como en la documentación 2. La vista tiene secciones con fuentes de información <ol style="list-style-type: none"> a. Función seleccionar fuente universidad b. Función seleccionar otra fuente 3. La vista tiene una caja de texto <ol style="list-style-type: none"> a. Función filtrar las noticias por el texto ingresado 4. La vista tiene una tabla con todas las noticias <ol style="list-style-type: none"> a. Selección de noticia de la tabla
<p>Resultados esperados (Que esperamos como resultado)</p> <ol style="list-style-type: none"> 1. Inspección visual 2. Inspección visual <ol style="list-style-type: none"> a. Despliega noticias fuente de la universidad b. Despliega noticias otra fuente 3. Inspección visual <ol style="list-style-type: none"> a. Muestra las noticias que concuerdan con el testo ingresado 4. Inspección visual <ol style="list-style-type: none"> a. Abre una nueva vista con la imagen ampliada y la información completa

Fuente: Elaborado por los autores

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: IOS-02
Tarea: Vista de servicios
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016
Descripción (que funcionalidad debe implementar para ser aceptada) <ol style="list-style-type: none">1. La vista de servicios debe mostrarse como en la documentación2. La vista tiene una colección de iconos uno por cada servicio<ol style="list-style-type: none">a. Función seleccionar icono servicio
Resultados esperados (Que esperamos como resultado) <ol style="list-style-type: none">1. Inspección visual2. Inspección visual<ol style="list-style-type: none">a. Despliega servicios disponibles en el portal web de la universidad

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: IOS-03
Tarea: Vista de Galería
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016
Descripción (que funcionalidad debe implementar para ser aceptada) <ol style="list-style-type: none">1. La vista de servicios debe mostrarse como en la documentación2. La vista tiene una colección de iconos uno por cada imagen o video<ol style="list-style-type: none">a. Función seleccionar icono servicio

<p>Resultados esperados (Que esperamos como resultado)</p> <ol style="list-style-type: none"> 1. Inspección visual 2. Inspección visual <ol style="list-style-type: none"> a. Reproduce el video o amplía la imagen

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: IOS-04
Tarea: Vista de Redes sociales
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016
<p>Descripción (que funcionalidad debe implementar para ser aceptada)</p> <ol style="list-style-type: none"> 1. La vista de redes sociales debe mostrarse como en la documentación 2. La vista tiene una colección de iconos uno por cada red social <ol style="list-style-type: none"> a. Función seleccionar icono de red social
<p>Resultados esperados (Que esperamos como resultado)</p> <ol style="list-style-type: none"> 1. Inspección visual 2. Inspección visual <ol style="list-style-type: none"> a. Despliega vista con un navegador que tiene direccionado al sitio web de la red social

HOJA DE PRUEBAS DE ACEPTACIÓN
Numero de historia: IOS-05
Tarea: Vista Información
Fecha de realización: 01 octubre 2016
Fecha de Ejecución: 02 de octubre 2016

<p>Descripción (que funcionalidad debe implementar para ser aceptada)</p> <ol style="list-style-type: none"> 1. La vista de información debe mostrarse como en la documentación 2. La vista tiene un contenedor para un mapa 3. Etiquetas de texto con información dirección 4. Etiqueta de texto con información de números de teléfono 5. Etiqueta de texto que muestra la información de la historia de la universidad
<p>Resultados esperados (Que esperamos como resultado)</p> <ol style="list-style-type: none"> 6. Inspección visual 7. Inspección visual 8. Inspección visual 9. Inspección visual 10. Inspección visual

Revisión de la Prueba de Aceptación.

Con los conocimientos adquiridos en base a los requisitos del cliente, las pruebas de aceptación y las tareas definidas difundimos este conocimiento a todos los miembros del equipo para que puedan sugerir posibles mejoras.

Día de trabajo

Envolver. Es una sesión interactiva para comunicar el progreso y problemas dentro del equipo. La conclusión suele realizarse como la primera y / o la última actividad de la jornada de trabajo (resumen de la mañana y Tarde de recapitulación).

Desarrollo Impulsado por Pruebas. En TDD las pruebas unitarias se escriben antes del código de programa. El código de programa se desarrolla para trabajar con las pruebas ya escritas. El propósito TDD es dar a los desarrolladores confianza que el código que producen las obras y guían el diseño del código para clarificar estructura más fácilmente.

Integración continúa. El propósito de la integración continua es integrar continuamente el nuevo código con el código existente en un código repositorio. Al integrar continuamente las integraciones masivas se pueden evitar que de lo contrario tomar mucho tiempo y esfuerzo.

La refactorización es el proceso de mejorar la capacidad interna de la estructura sin modificar su comportamiento externo. Con pequeñas mejoras en el código, la refactorización asegura que el software sea más modificable, extensible y legible.

Informar al Cliente. El objetivo de esta tarea es proporcionar una vista del progreso al cliente, y dar al cliente una posibilidad de dar retroalimentación sobre las características implementadas y guiar el desarrollo.

Día de liberación

Integración del sistema

En la presente iteración es la primera en ser desarrollada por lo que no es necesario la integración con el sistema o proyecto final. Más bien se realizan pruebas de funcionamiento unitario de esta iteración.

Resolución de problemas

A continuación de las pruebas unitarias determinamos si existen problemas o fallas que necesitan ser resueltas.

Reunión de entrega

Tabla 38 Verificación de auditorías

Lista de Verificación de auditorías			
fecha:	liberación #		
Requerimientos	SI	No	NA
1. todas las historias tienen el estado de "verificado"	X		
2. Todas las tareas tienen el estado de "verificado"	X		

3. Se han realizado pruebas previas a la liberación (y se han documentado problemas)	X		
4. Se han realizado pruebas de liberación (y se han documentado problemas)	X		
5. Los documentos están al día (revisión)	X		
6. Las métricas están actualizadas (bug-list, LOCs ...)	X		
7. El repositorio de control de versiones está actualizado (se han registrado todos los cambios relevantes)	X		
Comentarios: La iteración con el dispositivo móvil iPhone cumple con todos los requerimientos establecidos por el cliente			
seleccione uno: <input type="checkbox"/> Los resultados revisados satisfacen los requisitos y son aceptados. <input type="checkbox"/> Los resultados revisados son aceptados con cambios (Ver listas de resumen de deficiencia, página 2). <input type="checkbox"/> Los resultados revisados no cumplen con los requisitos (ver listas de resumen de deficiencia, página 2)			
Aprobado por: _____ Fecha: 31 de enero 2017			

Fuente: Elaborado por los autores

Estabilización

Día de planificación



Figura 38 Día de planificación

Fuente: Elaborado por los autores

Taller de post-iteración. El propósito del Taller de Post-iteración Es mejorar iterativamente el proceso de desarrollo de software para encajar mejor las necesidades del actual equipo de proyecto de software. Esto incluye la identificación de fortalezas y debilidades en proceso, así como la generación de acciones de mejora para la siguiente iteración.

Análisis de requerimientos. El propósito del Análisis de Requisitos es priorizar y analizar cuidadosamente los requisitos seleccionados para cada iteración. Durante esta tarea se garantiza que los requisitos que la mayoría de los valores de negocio se identifican, y esos requisitos son correctamente entendidos.

Planificación de la iteración. El objetivo del Plan de Iteración es generar la programación y el contenido de la iteración a ejecutar. El contenido es definido en términos de tareas que son órdenes de trabajo para el equipo.

Generación de Pruebas de Aceptación. El propósito de las Pruebas de Aceptación es verificar que los requisitos que el cliente ha establecido para el software correctamente. Las pruebas de aceptación se generan el día de la planificación. Posteriormente, las pruebas de aceptación se ejecutan del día de lanzamiento con el cliente y los problemas encontrados (es decir, defectos y mejoras) son documentados.

Revisión de la Prueba de Aceptación. El propósito de esta tarea es difundir conocimiento de las pruebas de Aceptación y los requisitos del sistema a todo el Equipo y permitir que los miembros del equipo comenten sobre cómo mejorar su calidad

Día de trabajo

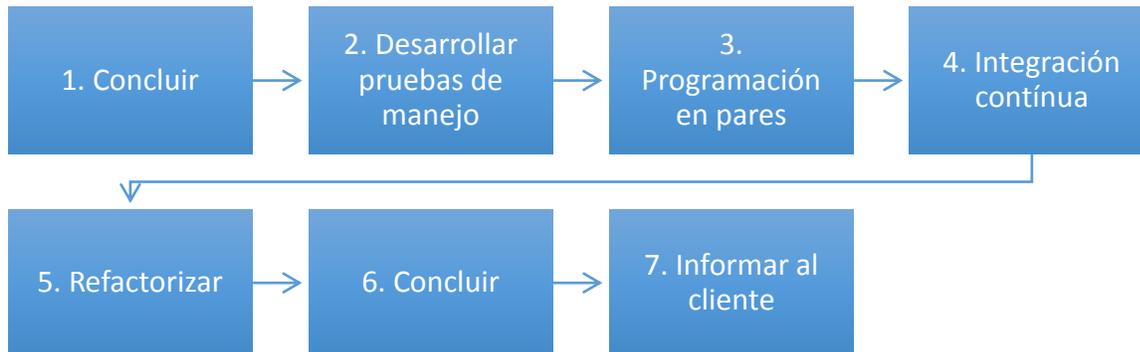


Figura 39 Día de trabajo

Fuente: Elaborado por los autores

Envolver. Es una sesión interactiva para comunicar el progreso y problemas dentro del equipo. La conclusión suele realizarse como la primera y / o la última actividad de la jornada de trabajo (resumen de la mañana y Tarde de recapitulación).

Desarrollo Impulsado por Pruebas. En TDD las pruebas unitarias se escriben antes del código de programa. El código de programa se desarrolla para trabajar con las pruebas ya escritas. El propósito TDD es dar a los desarrolladores confianza que el código que producen las obras y guían el diseño del código para clarificar estructura más fácilmente.

La programación de pares es un estilo de programación donde dos desarrolladores codifican juntos. El conductor es la persona que usa el teclado y el ratón durante una sesión de programación de pares. Navegador se sienta delante del computador al lado del conductor. El propósito de la programación de pares es mejorar comunicación, difundir el conocimiento dentro del equipo y asegurar la calidad del código.

Integración continua. El propósito de la integración continua es integrar continuamente el nuevo código con el código existente en un código repositorio. Al integrar continuamente las integraciones masivas se pueden evitar que de lo contrario tomar mucho tiempo y esfuerzo.

La refactorización es el proceso de mejorar la capacidad interna de la estructura sin modificar su comportamiento externo. Con pequeñas mejoras en el código, la refactorización asegura que el software sea más modificable, extensible y legible.

Informar al Cliente. El objetivo de esta tarea es proporcionar una vista del progreso al cliente, y dar al cliente una posibilidad de dar retroalimentación sobre las características implementadas y guiar el desarrollo.

Documentación

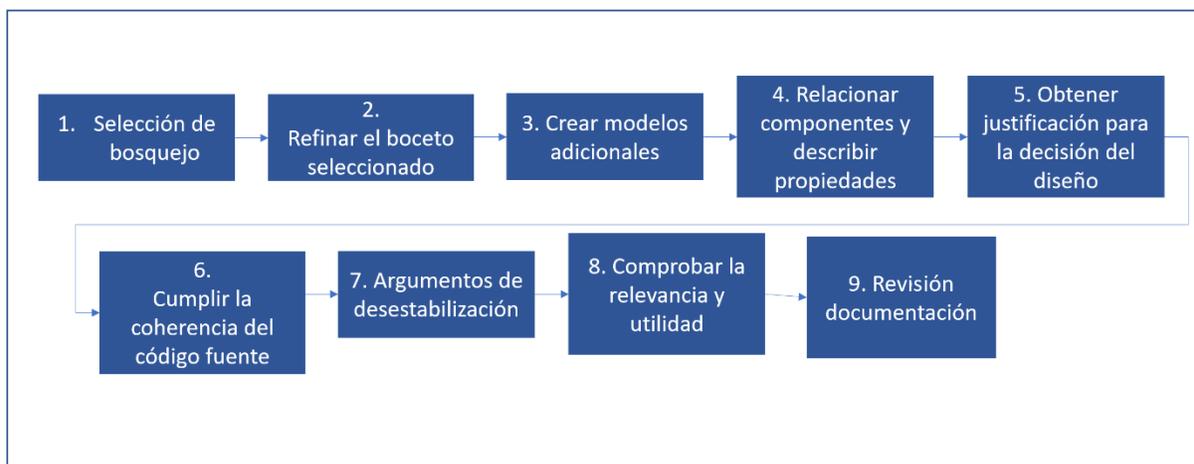


Figura 40 Documentación

Fuente: Elaborado por los autores

Seleccione los bocetos de arquitectura, diseño e interfaz de usuario que se refinarán y incluirán en los documentos y descartar los bocetos que han cumplido su propósito. Algunos de los bocetos pueden haber sido modelados en documentos anteriores.

Refinar y transformar los bocetos seleccionados en completos y consistentes modelos que pueden ser fácilmente entendidos por otras personas además del equipo del proyecto.

Cree modelos adicionales cuando sea necesario.

Describir las propiedades externamente visibles de los componentes en los modelos, y las relaciones entre ellos utilizando patrones arquitectónicos y de diseño cuando sea posible.

Recopilar, refinar e incluir la justificación de las decisiones de diseño de Arquitectura, diseño y documentos de interfaz de usuario. Utilizar patrones de arquitectura y diseño cuando sea posible.

Hacer cumplir que la documentación sea coherente con el código fuente y sus comentarios.

Aumentar los documentos de arquitectura, diseño y UI para que puedan transmitir información arquitectural, de diseño y de interfaz de usuario sobre el software desarrollado en una manera completa, consistente y formal para que otras personas que no sean del equipo del proyecto puede entender fácilmente lo que fue construido, por qué se construyó de esa forma y cómo operarlo.

Compruebe que la arquitectura, el diseño y los documentos de la interfaz de usuario sean cortos y útiles.

Revise los documentos de arquitectura, diseño y UI según el plan de proyecto.

Día de liberación

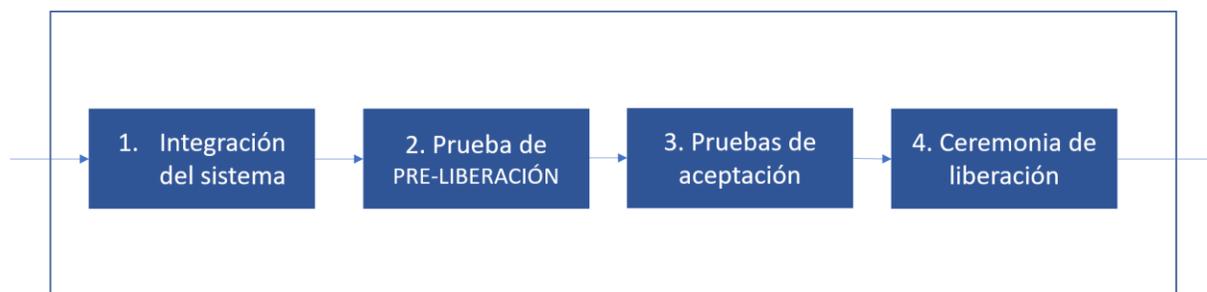


Figura 41 Día de liberación

Fuente: Elaborado por los autores

Integración de sistema. En el caso del proyecto multi-equipo, el propósito de esta tarea consiste en integrar subsistemas, generados en equipos separados, en un solo producto.

Pruebas de Pre-Liberación. El propósito de esta tarea es asegurarse de que el software que se está produciendo está listo para la Prueba de Aceptación y la liberación.

Las Pruebas Previas a la Liberación se llevan a cabo dentro del equipo ejecutando todas las pruebas de aceptación.

Prueba de aceptación. El propósito de las Pruebas de Aceptación es verificar que ellos requisitos que el cliente ha establecido para el software se implementan correctamente. Durante el día de lanzamiento, los ensayos de aceptación se ejecutan

Clientes y problemas encontrados (es decir, defectos y mejoras) están documentados.

Las ceremonias de lanzamiento son las tareas finales antes de hacer un lanzamiento de software. En la práctica, las ceremonias de liberación consisten en dos actividades esenciales;

La auditoría de lanzamiento y la creación de línea de base. Estas actividades se realizan para confirmar que todo ha sido hecho correctamente en la iteración actual y la base para garantiza un mayor desarrollo.

Pruebas y arreglos del sistema

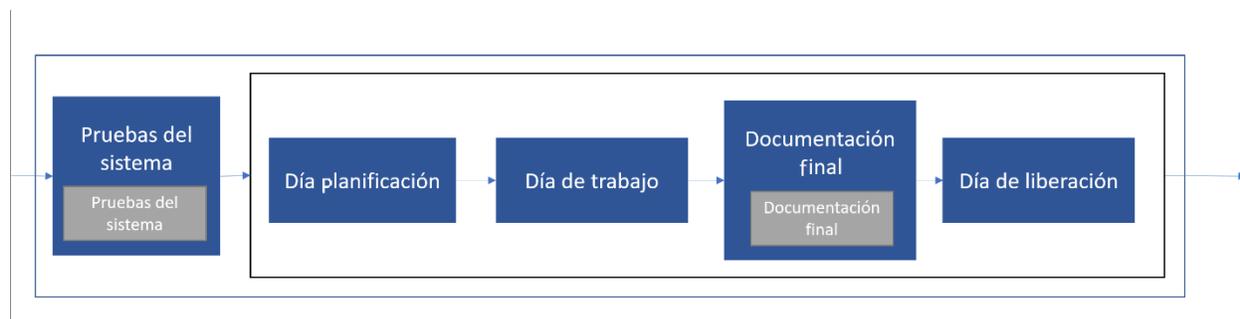


Figura 42 Pruebas y arreglos del sistema

Fuente: Elaborado por los autores

Pruebas del sistema

Es una etapa en la que el sistema se prueba según se describe en el patrón de tareas de prueba del sistema. Los defectos encontrados están documentados para el propósito de la iteración arreglar.

Pruebas del Sistema:

Modelo de teléfono:	- Android s6	- iPhone 7 plus
Software versión:	- 5.0	- 10.3
Nombre de la vista	Vista de inicio	

Encargado de las pruebas:	Diego Román	Marcelo Secaira
Criterio de las pruebas:	<ol style="list-style-type: none"> 1. ingreso a menú principal 2. validación por token 3. consulta y despliegue de noticias 4. consulta y despliegue de servicios 5. vista y reproducción de elementos de galería 6. vista y enlace a redes sociales 7. publicación de información 	
aceptada:	<ol style="list-style-type: none"> 1. SI 2. SI 3. SI 4. SI 5. SI 6. SI 	<ol style="list-style-type: none"> 1. SI 2. SI 3. SI 4. SI 5. SI 6. SI
Severidad:	moderada	moderada
Reproducible:		

Arreglar

Es una variación de la iteración normal; Sin embargo, no hay nueva funcionalidad y la escala de tiempo puede ser notablemente más corta. La entrada para esta iteración son los defectos encontrados en la etapa de prueba del sistema.

Planificación diaria

Día de la planificación. El objetivo de la etapa de planificación de la prueba del sistema & fase de arreglo es definir el contenido (es decir, historias y tareas) para la iteración de arreglo. Los defectos encontrados en la etapa de prueba del sistema son entradas para las descripciones de tarea.

Día de trabajo

El objetivo de la etapa de día de trabajo de la fase de prueba y arreglo del sistema es solucionar los defectos encontrados en la etapa de prueba del sistema y finalizar la implementación del producto.

Documentación

El objetivo de la Documentación es finalizar la arquitectura de software, diseño y documentos de interfaz de usuario. La documentación se actualiza para corresponder con los cambios realizados durante la iteración Arreglo. Como se muestra en la figura, la etapa de Documentación de Conclusión incluye una sola tarea, la documentación, que puede llevarse a cabo utilizando el patrón de tarea de conclusiones y de documentación.

Día de liberación

El propósito de la etapa Día de liberación es verificar y validar la funcionalidad implementada y la calidad de todo el software y su documentación. El día de lanzamiento culmina en la versión final de todo el software.

SECCIÓN IV

CONCLUSIONES

- Para desarrollar aplicaciones móviles de las plataformas IOS y Android, fue necesario entender el funcionamiento, lenguajes de programación, medios de comunicación, metodologías de desarrollo y líneas de arquitectura las cuales nos permitieron ampliar los conocimientos para futuros desarrollos que serán útiles en nuestro campo como profesionales.
- El desarrollo de la consola web de administración permitió manejar de manera ágil y dinámica la información de cada sección en las aplicaciones móviles.
- La implementación de los servicios web REST permitió interconectar las plataformas web y móviles utilizando la misma interfaz de comunicación.
- Para cumplir el propósito de este proyecto, se implementan las aplicaciones móviles en las tiendas virtuales PLAY STORE y APP STORE, de esta manera los estudiantes tienen la facilidad de acceder a las noticias que genera la Universidad y a sus servicios estudiantiles.

RECOMENDACIONES

- Para el correcto funcionamiento de las aplicaciones móviles, es necesario disponer de acceso a internet por medio de plan de datos o conectado a una red inalámbrica WIFI.
- Para la visualización de notificaciones es necesario aceptar los permisos para la recepción de las mismas.
- Los servicios universitarios agregados desde la consola web de administración, deberán estar programados utilizando técnicas de diseño responsivo o adaptivo, para que se visualicen de manera correcta en las aplicaciones móviles.
- El desarrollo de aplicaciones móviles se encuentra en constante evolución y se proyecta como una tendencia para los próximos años por lo que se sugiere continuar con el desarrollo de aplicación de la Universidad Tecnológica Israel implementando mejoras o agregando nuevas funcionalidades.

BIBLIOGRAFÍA

- Ableson, F., & Sen, R. (2009). *Unlocking Android*. (M. P. C.O., Ed.)
- Andersin, H., Niemi, E., & Hirvonen, V. (2005). *ICAM 2005: Proceedings of the International Conference on Agility*. Finland, Otaniemi: Helsinki University of Technology.
- Business Dictionary. (2017). *businessdictionary*. Retrieved Febrero 15, 2017, from [www.businessdictionary.com: http://www.businessdictionary.com/definition/exploratory-research.html](http://www.businessdictionary.com/definition/exploratory-research.html)
- Chung, C. (2008, Abril). <http://carloshungr.blogspot.com/>. Retrieved Enero 30, 2017, from [carloshungr.blogspot: http://carloshungr.blogspot.com/2008/04/enfoques-cuantitativo-deductivo-y.html](http://carloshungr.blogspot.com/2008/04/enfoques-cuantitativo-deductivo-y.html)
- Conder, S., & Darcey, L. (2009). *Android Wireless Application Development*. Reading, Massachusetts, United States of America: Addison-Wesley.
- Garcia, I., & Munilla, E. (2004). *E-BUSINESS COLABORATIVO*. Madrid, ESPANA: FC EDITORIAL.
- Google Inc. (2017). *developer.android.com*. Retrieved Enero 31, 2017, from [developer.android.com: https://developer.android.com/studio/intro/index.html](https://developer.android.com/studio/intro/index.html)
- Hall, S., & Anderson, E. (2009, Diciembre). Operating Systems for Mobile Computing. *Journal Of Computing Sciences in Colleges*.
<http://www.adwe.es/general/colaboraciones/servicios-web-restful-con-http-parte-i-introduccion-y-bases-teoricas>. (2011). Retrieved Febrero 13, 2017, from <http://www.adwe.es/general/colaboraciones/servicios-web-restful-con-http-parte-i-introduccion-y-bases-teoricas>
- <http://www.google.com/apis/>. (2011). Retrieved Febrero 13, 2017, from <http://www.google.com/apis/>
- <http://www.w3.org/2002/ws/soap12-mtom/>. (2011). Retrieved Febrero 13, 2017, from <http://www.w3.org/2002/ws/soap12-mtom/>
- <http://www.w3.org/TR/soap12-mtom/>. (2011, Mayo 08). Retrieved Febrero 13, 2017, from <http://www.w3.org/TR/soap12-mtom/>
- <http://www.w3.org/TR/ws-cdl-10/>. (2011, Mayo 08). Retrieved Febrero 13, 2017, from <http://www.w3.org/TR/ws-cdl-10/>

<http://www.w3.org/TR/wsd120/>. (2011, Mayo 08). Retrieved Febrero 13, 2017, from <http://www.w3.org/TR/wsd120/>

Lamarca, M. (2013, Diciembre 08). http://www.hipertexto.info/documentos/serv_web.htm. Retrieved Febrero 13, 2017, from http://www.hipertexto.info/documentos/serv_web.htm

McAllister, N. (n.d.). *Adroid Malware: How open is too open?* Retrieved Febrero 05, 2017, from Infowold: <http://www.infoworld.com/d/developer8world/android-malware-how-open-too-open-784>

Oracle Corporation. (2017, 02 15). www.java.com. Retrieved from java: https://www.java.com/en/download/faq/whatis_java.xml

Spataru, A. C. (2010). *Agile Development Methods for Mobile Applications*. Retrieved Febrero 10, 2017, from <http://www.inf.ed.ac.uk/publications/thesis/online/IM100767.pdf>

techtarget. (2017). *searchmobilecomputing*. Retrieved Enero 31, 2017, from <http://searchmobilecomputing.techtarget.com>: <http://searchmobilecomputing.techtarget.com/definition/iOS>

wikipedia. (2017). Retrieved febrero 13, 2017, from <https://es.wikipedia.org/wiki/Xcode>