

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

TEMA:

**ESTUDIO DE LA HERRAMIENTA IBM WORKLIGHT STUDIO PARA EL
DESARROLLO DE APLICACIONES MÓVILES MULTIPLATAFORMA CON
APLICATIVO DE CONSULTA DE NOTAS, LIBROS Y SÍLABOS EN EL
INSTITUTO TECNOLÓGICO SUPERIOR “JOSÉ CHIRIBOGA GRIJALVA”**

AUTOR:

VADIN SANTIAGO VILLOTA HIGUERA

DIRECTOR:

Ing. Diego J. Trejo Msc

Ibarra – Ecuador

2015

AUTORIZACIÓN DE USO DE PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

I. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determina la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto y pongo a disposición la siguiente información:

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD	100239331-0
NOMBRES:	VADIN SANTIAGO
APELLIDOS:	VILLOTA HIGUERA
DIRECCIÓN:	Isla Santa Isabel 20-28 y Machala – IBARRA
EMAIL:	vsvillota@utn.edu.ec
TELÉFONO FIJO:	062605790
TELÉFONO MÓVIL:	0992412986

DATOS DE LA OBRA	
TÍTULO:	ESTUDIO DE LA HERRAMIENTA IBM WORKLIGHT STUDIO PARA EL DESARROLLO DE APLICACIONES MÓVILES MULTIPLATAFORMA CON APLICATIVO DE CONSULTA DE NOTAS, LIBROS Y SÍLABOS EN EL INSTITUTO TECNOLÓGICO SUPERIOR “JOSÉ CHIRIBOGA GRIJALVA”
AUTOR(ES):	VADIN SANTIAGO VILLOTA HIGUERA

FECHA: AAAA-MM-DD	2015-01-22
SÓLO PARA TRABAJOS DE GRADO	
PROGRAMA:	PREGRADO
TÍTULO POR EL QUE OPTA	INGENIERO EN SISTEMAS COMPUTACIONALES
ASESOR DIRECTOR	ING. DIEGO J. TREJO MSC.

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Vadín Santiago Villota Higuera, con cédula de identidad Nro. 1002393310, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación del trabajo en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

3. CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos del autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 17 días del mes de Abril del 2015.

AUTOR

Vadín Santiago Villota Higuera

C.C.: 1002393310

Facultado por resolución del Consejo Universitario _____

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR

Yo, Vadin Santiago Villota Higuera con cédula de identidad Nro. 1002393310, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5, 6, en calidad de autor del trabajo de grado denominado **ESTUDIO DE LA HERRAMIENTA IBM WORKLIGHT STUDIO PARA EL DESARROLLO DE APLICACIONES MÓVILES MULTIPLATAFORMA CON APLICATIVO DE CONSULTA DE NOTAS, LIBROS Y SÍLABOS EN EL INSTITUTO TECNOLÓGICO SUPERIOR “JOSÉ CHIRIBOGA GRIJALVA”**, que ha sido desarrollado para optar por el título del Ingeniero en Sistemas Computacionales, en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento de hago entrega del trabajo final en el formato impreso y digital a la Dirección de Desarrollo Tecnológico e Informática de la Universidad Técnica del Norte.

(Firma):.....

Nombre: Vadin Santiago Villota Higuera

Cédula: 1002393310

Ibarra, a los 17 días del mes de Abril del 2015

CERTIFICACIÓN

El señor Vadin Santiago Villota Higuera ha trabajado en el desarrollo del proyecto de grado **ESTUDIO DE LA HERRAMIENTA IBM WORKLIGHT STUDIO PARA EL DESARROLLO DE APLICACIONES MÓVILES MULTIPLATAFORMA CON APLICATIVO DE CONSULTA DE NOTAS, LIBROS Y SÍLABOS EN EL INSTITUTO TECNOLÓGICO SUPERIOR “JOSÉ CHIRIBOGA GRIJALVA”**, previo a la obtención del Título de Ingeniero en Sistemas Computacionales, realizándola con interés profesional y responsabilidad, que certifico en honor a la verdad.



Ing. Diego J. Trejo MSc.

DIRECTOR DE TRABAJO DE GRADO



INSTITUTO TECNOLÓGICO SUPERIOR

“José Chiriboga Grijalva”

Ibarra-Imbabura
El Oro y 13 de Abril
Telf: 2558378 25450
www.tecnologicoitca.edu

CERTIFICADO

La Suscrita, Rectora del Instituto Tecnológico Superior “José Chiriboga Grijalva” con Registro Institucional 10-005 CONESUP CERTIFICA:

Que el Señor **VADIN SANTIAGO VILLOTA HIGUERA**, portador de la cédula de ciudadanía N°. 100239331-0, estudiante de la Universidad Técnica del Norte, realizó el aplicativo de la tesis con el tema: **ESTUDIO DE LA HERRAMIENTA IBM WORKLIGHT STUDIO, PARA EL DESARROLLO DE APLICACIONES MOVILES MULTIPLATAFORMA CON APLICATIVO DE CONSULTA DE NOTAS, LIBROS Y SÍLABOS EN EL INSTITUTO TECNOLÓGICO SUPERIOR “JOSÉ CHIRIBOGA GRIJALVA”** el mismo que hace entrega al Instituto Tecnológico Superior “José Chiriboga Grijalva” que ha sido revisado por el personal del departamento de sistemas y se ha determinado que cumple con los requerimientos técnicos solicitados.

El interesado puede hacer uso del presente certificado en la forma que creyere conveniente.

Ibarra, 2015-03-23

Atentamente

Dra. Alicia Soto M.; Esp
RECTORA



DEDICATORIA

Este trabajo de grado le dedico primeramente a Dios quien me dio la vida, también a mi amada madre, a mi querida y amorosa esposa y por ultimo a mi hermosa hija Kathleen.

AGRADECIMIENTOS

Agradezco primeramente a Dios quien cada día me da vida, y me demuestra que siempre está conmigo en todo momento, sé que puede contar con él siempre. Gracias Señor por guiar mi camino y perdonarme a pesar de los errores que cometo y por amarme mucho.

También quiero agradecer a mi madre, por apoyarme en mis estudios y cuidar de mí desde niño y darme su amor de madre, que sé que siempre lo voy a tener. Gracias mami por enseñarme a ser un buen hombre y temeroso de Dios, él le bendiga siempre.

También agradezco a mí amada esposa por haber estado siempre a mi lado en el transcurso de mis estudios, por haberme ayudado en esta etapa tan importante de mi carrera. Te agradezco mi amor por enseñarme los caminos de Dios y amarme mucho como sé que siempre lo harás.

Agradezco a mi hija Kathleen por ser cada día la felicidad y alegría mía y la de mí esposa. Mi hijita papi te ama mucho y estará contigo siempre.

Y por último también quiero agradecer al Ingeniero Diego Trejo que me ayudo en gran manera en el desarrollo de mi trabajo de grado, y que durante un periodo me dio su confianza para poder trabajar con él. Gracias inge por su apoyo y confianza.

En general quiero agradecer a mis docentes, que me supieron impartir grandes conocimientos y habilidad. Gracias Inges por sus palabras y conocimiento.

RESUMEN

El desarrollo de aplicaciones móviles ha aumentado con gran auge en estos tiempos por lo que es necesario que como desarrolladores de software se investiguen nuevas herramientas que permitan agilizar el desarrollo.

La empresa IBM una de las pioneras en tecnología de desarrollo ha incursionado en el desarrollo móvil con su producto IBM Worklight Foundation que es una Suite completa de herramientas de desarrollo móvil, dentro de una de esas herramientas se encuentra IBM Worklight Studio un componente que posee muchas ventajas para el rápido desarrollo de aplicaciones móviles y de escritorio, su principal característica es el desarrollo multiplataforma, esto quiere decir que a la vez que se va programando para un sistema operativo móvil también se puede programar para otro a la vez, esta como otras ventajas más hace de IBM Worklight Studio una herramienta digna de estudiar a profundidad y de conocimiento para los estudiantes de esta prestigiosa Universidad.

Este trabajo de grado tiene como finalidad realizar un estudio completo de la herramienta de desarrollo móvil IBM Worklight Studio, y demostrar sus cualidades mediante una aplicación web para una Institución de educación superior.

En el primer capítulo se describe la necesidad de investigar nuevas herramientas en el campo de desarrollo móvil para la formación de ingenieros y desarrolladores de software.

En el segundo capítulo se describe en general todos los conceptos, herramientas, plataformas, sistemas operativos que se poseen dentro del campo de desarrollo móvil.

En el tercer capítulo se describe toda la investigación sobre la herramienta IBM Worklight Studio como también se habla un poco acerca de la Suite de la cual forma parte.

En cuarto capítulo se describe el desarrollo de la aplicación móvil para consulta de sílabos, bibliografía y notas aplicando la metodología SCRUM para su desarrollo.

En el quinto capítulo se describe las conclusiones y recomendaciones.

SUMMARY

The mobile application development has increased with successful these days, so it is necessary as software developers' new tools would be good to be developed that let speediness this research.

The IBM company a pioneer in developing technology has dip into in mobile development with IBM Worklight Foundation which is a complete suite of tools for mobile development within one of those tools IBM Worklight Studio is a component that has many advantages for the rapid development of mobile and desktop applications, its main feature is the platform development, this means that at a time to be programmed for a mobile operating system can also be programmed to another at the same time, this and other advantages more IBM Worklight Studio makes a worthy tool to study in depth and knowledge to the students of this prestigious university.

This work aims grade a full study of mobile development tool IBM Worklight Studio, and demonstrate its qualities through a web application to an institution of higher education.

In the first chapter the need to investigate new tools in the field of mobile development for the training of engineers and software creators described.

In the second chapter describes in general all the concepts, tools, platforms, operating systems have within the field of mobile growth.

In the third chapter all research on the IBM Worklight Studio tool is described as also talks a bit about the Suite which is a part.

In the fourth chapter the development of mobile application to query syllabus, bibliography and notes using the SCRUM methodology for its development are described.

In the fifth chapter the conclusions and recommendations described.

CONTENIDO

1. INTRODUCCIÓN.....	1
1.1. ANTECEDENTES	1
1.1.1. Descripción.....	1
1.2. SITUACIÓN	1
1.3. PROBLEMA	1
1.4. JUSTIFICACIÓN	2
1.4.1. Impactos	2
1.5. OBJETIVOS	3
1.5.1. Objetivo General.....	3
1.5.2. Objetivos Específicos	3
1.6. ALCANCE.....	3
2. MARCO TEÓRICO	5
2.1. FUNDAMENTACIÓN TEÓRICA	5
2.1.1. Evolución de los Sistemas Operativos	5
2.1.2. Sistemas Operativos de Dispositivos Móviles	7
2.2. DESARROLLO DE APLICACIONES MÓVILES	12
2.2.1. Desarrollo para Android	13
2.2.2. Desarrollo para iOS	17
2.2.3. Desarrollo para Windows Phone	20
2.3. TIPO DE APLICACIONES MÓVILES.....	21
2.3.1. Aplicaciones Web.....	21
2.3.2. Aplicaciones Nativas	23
2.3.3. Aplicaciones Híbridas.....	24
2.4. HERRAMIENTAS DE DESARROLLO MÓVIL MULTIPLATAFORMA	25
2.4.1. IBM Worklight	25
2.4.2. PhoneGap	26

2.5. METODOLOGÍA SCRUM.....	27
MARCO INVESTIGATIVO.....	30
3. IBM WORKLIGHT STUDIO.....	30
3.1. IBM WORKLIGHT FOUNDATION	30
3.1.1. Novedades de la Herramienta.....	32
3.2. CARACTERÍSTICAS DE IBM WORKLIGHT STUDIO	33
3.2.1. Soporte Multiplataforma	33
3.2.2. Codificación Híbrida	34
3.2.3. Runtime Skin	35
3.2.4. Soporte para HTML5	36
3.2.5. Integración del SDK de dispositivo nativo.....	37
3.2.6. Recuperación estandarizada de datos	37
3.2.7. Notificaciones Push unificadas.....	38
3.3. COMPONENTES	39
3.3.1. Project.....	39
3.3.2. Adapters.....	42
3.3.3. ENVIROMENT	45
3.3.4. SKINS.....	47
3.4. HERRAMIENTAS	51
3.4.1. Rich Page Editor.....	51
3.4.2. Mobile Browser Simulator	52
3.5. DESARROLLO DE APLICACIONES	53
3.5.1. Proceso de desarrollo de una aplicación móvil	54
3.5.2. Desarrollo de aplicaciones web e híbridadas	55
3.5.3. Desarrollo de aplicaciones nativas	60
3.6. OPTIMIZACIÓN Y ACELERACIÓN DE COMPONENTES.....	63
3.6.1. Aceleración de componentes	63

3.6.2.	Optimización de componentes.....	65
3.7.	ADMINISTRACIÓN Y MONITOREO.....	71
4.	MARCO DE DESARROLLO.....	73
4.1.	GENERAL.....	73
4.1.1.	Introducción.....	73
4.1.2.	Propósito.....	73
4.1.3.	Alcance	73
4.1.4.	Generalidades	73
4.2.	COMPONENTES.....	74
4.2.1.	Roles y Responsabilidades	74
4.2.2.	Elementos	74
4.2.3.	Reuniones y Aplicación en el Proyecto.....	75
4.3.	DESARROLLO DE LA APLICACIÓN	88
4.3.1.	Arquitectura de la solución.....	88
4.3.2.	Base de datos	89
4.4.	INTERFAZ DE USUARIO	93
5.	CONCLUSIONES Y RECOMENDACIONES	98
5.1.	CONCLUSIONES	98
5.2.	RECOMENDACIONES.....	98
5.3.	GLOSARIO DE TÉRMINOS	99
5.4.	REFERENCIAS BIBLIOGRÁFICAS	100

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Módulo de Aplicativo Móvil de la Investigación.....	4
Ilustración 2 Línea del tiempo de Sistemas Operativos	7
Ilustración 3 SO Android.....	9
Ilustración 4 Nuevo SO para Smart phones Windows Phone7	9
Ilustración 5 Sistema Operativo Symbian	11
Ilustración 6 Evolución del Sistema Operativo iOS	12
Ilustración 7 Ejemplo de una clase Java.....	13
Ilustración 8 Ventana de Android Studio	15
Ilustración 9 Applications programs Eclipse.....	16
Ilustración 10 Objective-C tercer lenguaje más utilizado	17
Ilustración 11 Interoperabilidad a nivel de lenguaje	19
Ilustración 12 Aplicaciones Web.....	22
Ilustración 13 App nativas vs App Web.....	24
Ilustración 14 Métodos de desarrollo de aplicaciones móviles	24
Ilustración 15 Componentes de IBM Worklight	26
Ilustración 16 Diagrama del Proceso SCRUM.....	28
Ilustración 17 Componentes de Worklight.....	31
Ilustración 18 Elementos de un Proyecto de Android con IBM Worklight Studio.....	35
Ilustración 19 Escenarios del Runtime Skin.....	36
Ilustración 20 Infraestructura de Notificaciones	38
Ilustración 21 Proyecto Worklight	39
Ilustración 22 Estructura de un proyecto Worklight	41
Ilustración 23 Proyecto de Worklight Adapter.....	42
Ilustración 24 Comunicación base de datos con JSON y Worklight.....	43
Ilustración 25 Estructura de una adaptador de IBM Worklight.....	45
Ilustración 26 Estructura de un adaptador con función Javascript y XML	45
Ilustración 27 Proyecto de Worklight Environment.....	46
Ilustración 28 Creación de un Ambiente Worklight.....	47
Ilustración 29 Estructura generada de un Worklight Application Skin para Android.....	48
Ilustración 30 Método skinLoader.....	48
Ilustración 31 Representación gráfica básica de JSONStore	49
Ilustración 32 Componentes en interacción de JSONStore.....	50

Ilustración 33 Comparación entre tecnologías de almacenamiento de datos	50
Ilustración 34 Simulador Móvil de IBM Worklight	53
Ilustración 35 Desarrollo de una aplicación móvil	54
Ilustración 36 Anatomía de un Proyecto Worklight	55
Ilustración 37 Anatomía de una Aplicación Worklight	56
Ilustración 38 Controlador de eventos en Worklight	57
Ilustración 39 Componentes Metadata de la Aplicación	64
Ilustración 40 Estructura típica de una carpeta de un componente de aplicación	64
Ilustración 41 Estructura del archivo component.wcp	65
Ilustración 42 Propiedades de Minificación en Worklight	69
Ilustración 43 Arquitectura de la Aplicación Móvil	89
Ilustración 44 Tablas para extraer datos para las búsquedas bibliográficas	90
Ilustración 45 Tablas para extraer datos de estudiantes	91
Ilustración 46 Tablas para extraer datos de notas y materias	91
Ilustración 47 Tablas para extraer datos de los sílabos	92
Ilustración 48 Pantalla de Inicio de la aplicación	93
Ilustración 49 Pantalla de Logeo a la aplicación	94
Ilustración 50 Pantalla de menú del Usuario	94
Ilustración 51 Pantalla de lista de materias para el usuario	95
Ilustración 52 Pantalla de menú para consultar opciones académicas	95
Ilustración 53 Pantallas de Notas del usuario	96
Ilustración 54 Pantallas de Sílabos	96
Ilustración 55 Pantallas de Búsqueda Bibliográfica	97
Ilustración 56 Pantalla de Información de Búsqueda de Bibliografía	97

ÍNDICE DE TABLAS

Tabla 1 Vistas del Rich Page Editor	52
Tabla 2 Opciones para la minificación a nivel de atributo	69
Tabla 3 Tabla de Actores por Roles y Responsabilidades.....	74
Tabla 4 Primer Sprint de Desarrollo.....	76
Tabla 5 Primer Sprint Backlog de Desarrollo	76
Tabla 6 Primer Product Backlog de Desarrollo.....	77
Tabla 7 Segundo Sprint de Desarrollo.....	77
Tabla 8 Segundo Sprint Backlog de Desarrollo	78
Tabla 9 Segundo Product Backlog de Desarrollo.....	79
Tabla 10 Tercer Sprint de Desarrollo	79
Tabla 11 Tercer Sprint Backlog de Desarrollo.....	80
Tabla 12 Tercer Product Backlog de Desarrollo	81
Tabla 13 Cuarto Sprint de Desarrollo.....	82
Tabla 14 Cuarto Sprint Backlog de Desarrollo	83
Tabla 15 Cuarto Product Backlog de Desarrollo.....	84
Tabla 16 Quinto Sprint de Desarrollo.....	85
Tabla 17 Quinto Sprint Backlog de Desarrollo	86
Tabla 18 Quinto Product Backlog de Desarrollo	86
Tabla 19 Sexto Sprint de Desarrollo.....	87
Tabla 20 Sexto Sprint Backlog de Desarrollo	88
Tabla 21 Sexto Product Backlog de Desarrollo	88

1. INTRODUCCIÓN

1.1. ANTECEDENTES

1.1.1. Descripción

En el mundo entero cada vez va creciendo la industria de los dispositivos móviles, gracias a este auge cada vez se van creando aplicaciones móviles que tienen como objetivo brindar ayuda, entretenimiento, soporte, etc. Estas aplicaciones han permitido que se mantenga un contacto más cercano entre empresa y usuario, por lo que en la actualidad muchas empresas y personas desean realizar sus propias aplicaciones y ofrecer este servicio. En diversas empresas se han creado departamentos informáticos en los cuales se están creando software, estas están siendo dirigidas tanto para la web como también para celulares.

1.2. SITUACIÓN

En las empresas se están desarrollando aplicaciones dirigidas a dispositivos móviles, estas aplicaciones en su mayoría son creadas para dispositivos con diferentes sistemas operativos, por lo que cada programador se dedica en desarrollar sus aplicaciones dirigido únicamente para un sistema operativo, ya que no puede reutilizar el mismo código.

1.3. PROBLEMA

En el mundo de la programación se ha trabajado con herramientas de software de desarrollo móvil cuya realización es específicamente para un sistema operativo de dispositivos móviles, esto ha hecho que los programadores creen aplicaciones no escalables aumentando más tiempo en la creación de aplicaciones.

Actualmente se sigue trabajando de la misma forma, con la diferencia que cada vez se van creando plataformas para el desarrollo ágil de aplicaciones móviles multiplataforma. La empresa IBM¹ adquirió y mejoró un software que permite la programación multiplataforma con mayor escalabilidad, que facilita la creación, ejecución y gestión de aplicaciones

¹ **IBM:** International Business Machines, es una empresa norteamericana que desarrolla software para creación de software y hardware de diferentes tipos.

HTML5², híbridas y nativas; a su vez con la misma se podrá reducir costos en desarrollo y mantenimiento.

El desarrollo móvil multiplataforma es un nuevo nivel de programación en el mundo entero por todos los beneficios que se espera de la misma, es necesario que también en el Ecuador y en nuestros medios se realicen aplicaciones con este tipo de desarrollo. Sin el estudio de la Herramienta IBM Worklight Studio se negaría la oportunidad de que programadores conozcan sobre una alternativa para desarrollar aplicaciones móviles no solo para una plataforma sino para varias.

Los programadores no cuentan con un software para el desarrollo de aplicaciones móviles para varios sistemas operativos, la empresa IBM ha creado un software multiplataforma que permite realizar la programación dirigida a diversos SO³ móviles además de contar con muchos beneficios.

1.4. JUSTIFICACIÓN

Con el fin de justificar la siguiente investigación, se presenta el estudio de los impactos de la herramienta.

1.4.1. Impactos

- ❖ **Impactos Económicos:** se logrará en un futuro que los programadores quienes estén encargados de producir aplicaciones móviles, ya no gasten más tiempo y dinero programando para un solo sistema operativo, sino invertir en un producto que les permite desarrollar aplicaciones a nivel multiplataforma, lo que reducirá los costos de producción y los salarios por la contratación de empleados.
- ❖ **Impacto Tecnológico:** al realizar esta investigación se contribuirá al desarrollo de software dirigido a multiplataforma, ya que actualmente no se toman en cuenta las ventajas que esta posee, además no se tiene un conocimiento a profundidad del software multiplataforma como también de la herramienta IBM Worklight Studio⁴, por el hecho de ser un producto nuevo en su categoría.

² **HTML5:** es la quinta versión de HTML, que es un lenguaje de marcado de etiquetas, con el que se puede desarrollar aplicaciones web. La quinta versión posee nuevas y funcionales herramientas y etiquetas que ayudan a generar rápidamente el código.

³ **SO:** abreviación de Sistema Operativo

⁴ **IBM Worklight Studio:** es una de las herramientas que posee IBM Worklight, pero esta es un framework en el que se puede crear software dirigido para aplicaciones móviles a nivel multiplataforma.

❖ **Impacto Educativo y Social:** Lo que se pretende con el estudio de ésta herramienta es conocer a profundidad sus componentes, características y variedad de aplicaciones que ofrece, y verificar sus beneficios, con el fin de dar a conocer a los programadores de nuestro medio quienes recientemente practican el desarrollo de aplicaciones móviles, una nueva forma de programarlos.

Al realizar esta investigación se contribuirá al aprendizaje del desarrollo de programación multiplataforma dentro de las aulas de la Universidad Técnica del Norte, como también para los proyectos de tesis que se pueden presentar a futuro.

La metodología que se utilizará para el desarrollo de la aplicación es SCRUM⁵, ya que la aplicación móvil que se realizará es pequeña, necesita un desarrollo ágil.

1.5. OBJETIVOS

1.5.1. Objetivo General

Realizar el estudio de la herramienta de desarrollo IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma.

1.5.2. Objetivos Específicos

- Determinar la situación actual del desarrollo de aplicaciones móviles.
- Identificar herramientas de desarrollo móvil multiplataforma.
- Investigar sobre la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma.
- Desarrollar una aplicación móvil que demuestre la funcionalidad de la Herramienta IBM Worklight Studio en Instituto Tecnológico Superior “José Chiriboga Grijalva”.

1.6. ALCANCE

En el estudio de la herramienta se investigará los componentes y las características principales de la herramienta IBM Worklight Studio para desarrollar aplicaciones móviles.

La aplicación a desarrollar será una aplicación híbrida que contará con un módulo de 3 funciones principales y se realizarán las pruebas en un dispositivo Android.

⁵ **SCRUM:** es una metodología de desarrollo ágil de software, en el que interesa plasmar los requerimientos del usuario de manera rápida.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

- Consulta de notas del ciclo actual
- Consulta de libros en la biblioteca
- Consulta de Sílabos de materias

La demostración de la aplicación se la realizará en un dispositivo Android⁶, para las plataformas iOS⁷ y Windows Phone⁸ se las realizarán en un emulador.

En caso que el Instituto quisiera hacer pública la aplicación, se tendrá que comprar las licencias respectivas, pero estas se la realizarán por parte de la misma.

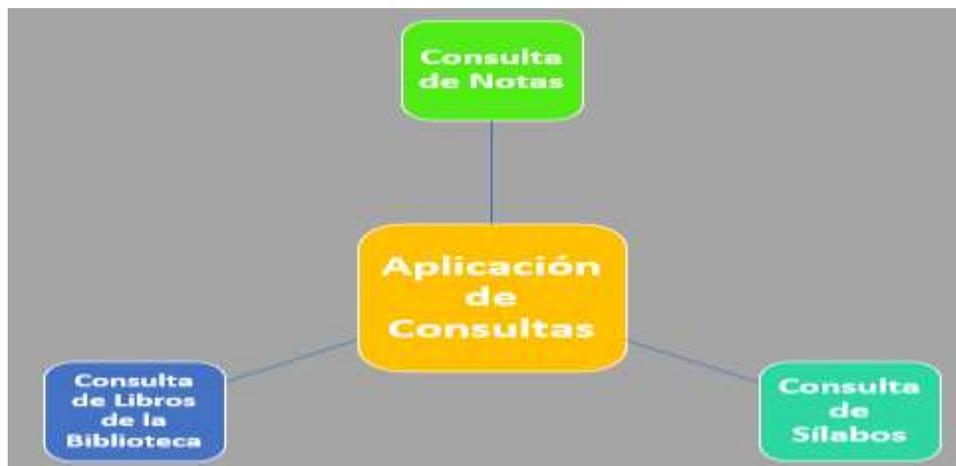


Ilustración 1 Módulo de Aplicativo Móvil de la Investigación

Fuente: Propia

⁶ **Android:** es un sistema operativo móvil creado por Google, su desarrollo es completamente libre.

⁷ **iOS:** iPhone Operating System, es un sistema operativo móvil creado por Apple.

⁸ **Windows Phone:** es un sistema operativo móvil desarrollado por Microsoft, como sucesor de la plataforma Windows Mobile.

2. MARCO TEÓRICO

2.1. FUNDAMENTACIÓN TEÓRICA

2.1.1. Evolución de los Sistemas Operativos

En la década de 1940, los primeros sistemas electrónicos digitales no tenían sistemas operativos. Los ordenadores de esas épocas eran tan primitivos comparados con los de hoy en día, los programas se introdujeron de a poco en los interruptores mecánicos.

Originalmente cada usuario escribía el código necesario para implementar una aplicación concreta, incluidas las instrucciones de entrada/salida; sin embargo muy rápidamente esta codificación se consolidó en un sistema de control IOCS⁹.

“Los primeros sistemas de almacenamiento se centraron al trabajo y no al tiempo de duración; el esfuerzo y tiempo que implicaba el montaje y desmontaje de las primeras máquinas almacenadoras de información, generó la necesidad de mejorar su eficiencia pensando en su automatización, siendo el primer sistema operativo el que fuera diseñado por el Laboratorio de Investigación de General Motors, para su arquitectura IBM 701, a principios de 1956”. (Castaño, 2014)

Entre los años sesenta y ochenta, se dieron muchos logros en los ámbitos de la informática y la computación; aparecieron técnicas como la multiprogramación y los sistemas de tiempo compartido, lo que implicaba dar un vuelco en los sistemas operativos; que generen interacción entre la máquina y el usuario, haciendo que el trabajo de procesamiento de información de la máquina sea más eficiente.

El año de los ochenta se caracteriza por el uso de los microprocesadores, el tener una computadora deja de ser inalcanzable, inclusive para personas sin vastos conocimientos técnicos de la informática; lanzándose al mercado variedad de ordenadores personales con sistemas operativos intuitivos, simples y amigables; lo que ocasionó el surgimiento de menús e interfaces gráficas con lenguajes de programación como: C, C⁺⁺, Haskell, Miranda, Eiffel y Smalltalk¹⁰.

⁹ IOCS: es un indicador de sistema de control.

¹⁰ C, C⁺⁺, Haskell, Miranda, Eiffel, Smalltalk son lenguajes de programación

A mediados de los ochenta se desarrollaron redes de computadoras personales con sistemas operativos en red y distribuidos, siendo MS-DOS y Unix los más usados¹¹.

En la década de los noventa con la aparición del Internet y el gran uso de las redes, hacen que los sistemas operativos estén diseñados para que el usuario acceda simultáneamente a estos dos servicios, generando motivación y expectativas; inclusive las aplicaciones multimedia empiezan a ser parte de los sistemas operativos, como también la flexibilidad y compatibilidad de accesorios de hardware con los cuales el computador puede interactuar, como cámaras fotográficas y de video, micrófonos, parlantes y los conocidos y muy utilizados USB.

En los años noventa, sobresale Linux un sistema operativo completamente libre basado en la plataforma UNIX, y es notorio lo que hace Microsoft en mejorar su sistema operativo Windows a través de sus varias versiones, Windows 3.0, Windows 3.1, Windows 95 y Windows 98. Es una época en la que la industria de la tecnología de la comunicación se siente atraída en emprender la fabricación de variedad de dispositivos móviles con sistemas operativos específicos, que satisfagan al consumidor, sin que este tenga ninguna complicación de uso, y pueda comunicarse a través de las redes sociales, a través de su correo electrónico, pueda captar o enviar fotos o cualquier tipo de información, dependiendo de los requerimientos personales o empresariales.

El nuevo milenio produjo grandes cambios en el mercado de los sistemas operativos, incrementando la competencia entre los diferentes desarrolladores, siendo Windows, Unix, y Mac OS los más competitivos, todos intentan satisfacer las necesidades de comunicación e interacción de sus clientes, desarrollando sistemas operativos para dispositivos cada vez más sofisticados con variedad de servicios, poniendo a su alcance lo que está de moda, lo inalámbrico, la comunicación en tiempo actual y el internet.

¹¹ MS-DOS, Unix son sistemas operativos de los años ochenta.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”



Ilustración 2 Línea del tiempo de Sistemas Operativos

Fuente: <http://sistemasoperativosvm.blogspot.com/2013/02/linea-del-tiempo-de-los-sistemas.html>

Actualmente existen sistemas operativos para dispositivos móviles muy sofisticados que permiten la interacción con el usuario, dejando atrás el uso del mouse y el teclado; el reconocimiento de voz, o la pantalla táctil es atractivo y marca el inicio de un avance mucho mayor en las tecnologías de la comunicación.

2.1.2. Sistemas Operativos de Dispositivos Móviles

“Los sistemas operativos son programas que se encargan de manejar los procesos básicos de un dispositivo, permitiendo el uso de sus diferentes recursos; inicialmente fueron desarrollados para computadoras, pero actualmente son utilizados para dispositivos móviles como celulares, tabletas electrónicas, para tener la misma interlocución entre el hardware y los programas que el usuario desea utilizar”. (Alto Nivel, 2013)

Es un gran programa hecho para controlar otros programas o aplicaciones, también controla las partes físicas de un dispositivo, imprescindible por ejemplo para tomar fotos, capturar videos, usar el teclado o ejecutar aplicaciones.

No todos los sistemas operativos son iguales, ni todos los dispositivos móviles son compatibles para todos, cada uno tiene sus características específicas, en

funcionalidad, compatibilidad, memoria, arquitectura, desarrollo y aplicación, depende del tipo y de la marca de celular, PDAs¹², consola o tableta electrónica.

Todos los dispositivos móviles emplean información electrónica y tienen una gran capacidad de procesamiento de sí mismos y de variedad de elementos multimedia, cuentan con elementos de captura de imágenes, capacidad de conexión alámbrica, inalámbrica y demás funciones; donde un sistema operativo es imprescindible para cumplir con su propósito de fabricación.

Entre algunos sistemas operativos para celulares se tiene: Android, iOS, Windows Phone, Blackberry OS y Symbian OS.

- **Sistema Operativo Android**

Es un sistema operativo dirigido para personas que buscan experiencias de comunicación y entretenimiento. Desarrollado por Google y la Open Handset Alliance, entre 2007- 2008, trabaja en una plataforma de código abierto; lo que da libertad al desarrollador, al fabricante y al usuario para personalizar el teléfono de acuerdo a sus gustos y necesidades.

Android está basado en Linux, fue diseñado originalmente para dispositivos móviles como los teléfonos inteligentes, pero después tuvo modificación para ser usado en tablets, actualmente se encuentra en desarrollo constante e incluso para utilizarse en netbooks y PCs.

Se destacan algunas ventajas de este sistema operativo, tales como:

- ✓ *“El código de Android es abierto, cualquier persona puede realizar una aplicación para este sistema operativo.*
- ✓ *Hoy en día hay más de 650 000 aplicaciones disponibles para teléfonos Android, aproximadamente 2/3 son gratis, está implantado en Tablets, GPS, relojes, microondas y más equipos, incluso hay por internet una versión para PC.*
- ✓ *El sistema Android es capaz de hacer funcionar a la vez muchas aplicaciones y además se encarga de gestionarlas, de dejarlas en modo de suspensión si*

¹² **PDAs:** es una agenda electrónica que por lo general emplean personas de negocios.

no se utilizan e incluso cerrarlas si llevan un largo periodo de inactividad, para evitar el consumo excesivo de batería”. (Scoello12, 2012)



Ilustración 3 SO Android

Fuente: www.lastdragon.net

Es una stack de software que incluye sistemas operativos Middleware y aplicaciones de base; los desarrolladores pueden crear aplicaciones para la plataforma usando el SDK de Android¹³, las solicitudes se han escrito utilizando el lenguaje de programación Java y se ejecutan en Dalvik¹⁴.

2.1.2.1. Sistema Operativo Windows Phone

Es un sistema operativo dirigido para las personas que buscan tener una oficina móvil además de todas las herramientas que brinda Microsoft Windows.



Ilustración 4 Nuevo SO para Smart phones Windows Phone7

Fuente: Webadicto.net

¹³ **SKD de Android** es un kit de desarrollo de software de este SO.

¹⁴ **Dalvik** es una máquina virtual del núcleo Linux.

Es un sistema operativo de mayor calidad que su antecesor el Windows Mobile que cuenta con una interfaz de usuario basada en el sistema de diseño de Windows Phone de Microsoft, inspirada en la interfaz de usuario Zune HD. Se caracteriza por presentar un mosaico dinámico con enlaces directos hacia aplicaciones, características, funciones y objetos individuales del teléfono como contactos, mensajes, páginas web, juegos, música, videos, otros, cada uno independiente y personalizable dependiendo de la necesidad del usuario; se encuentra disponible en los equipos móviles de sus principales socios comerciales HTC, HP, LG, Toshiba, Sony y Ericsson, Samsung, otros.

“Su interfaz se basa en baldosas dinámicas que ofrecen información desde la misma pantalla de inicio y cuyo tamaño se puede ajustar, prácticamente todo se puede convertir en baldosa, como una aplicación, un contacto, una lista de reproducción; la última versión Windows Phone 8,1 se caracteriza por esto, inclusive tiene un asistente de voz.

La pantalla de inicio en fondo negro es una gran ventaja, ya que permite a lo máximo el ahorro de batería, aunque el usuario puede cambiar el color de fondo a su gusto. Otro punto a favor es la sincronización casi automática con Skydrive, el servicio de almacenamiento en la nube de Microsoft, el cual ofrece almacenamiento gratis hasta de 25GB”. (Windows, 2013)

La sociedad actual es tan competitiva en todas las áreas, en el campo de la tecnología de la comunicación no se deja esperar, en el mercado de la telefonía y equipo móvil, existe mucha disputa entre las grandes empresas de sistemas operativos de dispositivos móviles; Microsoft en su dinamismo de innovación pretende prestar mejor servicio a sus usuarios, lanzando sus diferentes versiones conocidas WP7, WP8, y actualmente WP10.

2.1.2.2. Sistema Operativo Symbian

Está dirigido para personas que buscan un sistema operativo intuitivo y estable, además de un reproductor multimedia, cámara fotográfica, música y video, trabaja en una plataforma de código abierto, y lo desarrolla Symbian Foundation desde hace poco menos de 10 años.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

Fue diseñado exclusivamente para ser utilizado en móviles y por esta razón cuenta con algunas características que lo distingúan de los ya normales sistemas operativos; pero al igual que los otros soporta todo tipo de aplicaciones y permite navegar por internet en una interfaz totalmente personalizable, sin embargo la falta de desarrolladores y aplicaciones hicieron que quede casi relegado.



Ilustración 5 Sistema Operativo Symbian

Fuente: movilarena.com

Para la programación de aplicaciones para este sistema operativo no se requieren de conocimientos específicos de códigos, ya que se emplean lenguajes como Java, C++ Visual Basic, Python, Perl, Flash Lite y otros.

- **Sistema Operativo iOS**

iOS es un sistema operativo que da vida a dispositivos como el iPhone, iPad, iPod Touch o el Apple TV; su simplicidad y optimización son sus pilares para que millones de usuarios se sientan atraídos por iOS en lugar de escoger otras plataformas que necesitan más hardware para mover con fluidez el sistema.

“Su interfaz gráfica está diseñada para el touch screen, con capacidad para gestos multitouch, está constituida básicamente por sliders, interruptores y botones, con una respuesta fluida casi inmediata, soporta acelerómetros internos al dispositivo, emplea unos 500MB de almacenamiento, aunque varía según el modelo” (Dan, 2014).



Ilustración 6 Evolución del Sistema Operativo iOS

Fuente: globedia.com

Incluye múltiples aplicaciones para gestionar emails, fotos, cámara, mensajes, clima, notas, YouTube, contactos, reloj, y demás y soporta multitareas; no tiene soporte para Adobe Flash, ni Java, por tanto los sitios web en dicha tecnología no pueden ser vistos en este sistema operativo.

Apple está innovando cada año su stock es sus diferentes versiones, con el fin de satisfacer las múltiples necesidades personales y empresariales, en lo que se refiere a la comunicación, información, diversión, cálculo, música, video, otros.

2.2. DESARROLLO DE APLICACIONES MÓVILES

Dentro de las necesidades humanas en el campo de las tecnologías de la comunicación, se han incorporado un sinnúmero de requerimientos dependiendo del perfil del usuario, sea este un personaje público o privado; desde el espacio donde se desenvuelve hace uso de variedad de aplicaciones desarrolladas exclusivamente para dispositivos móviles, para actividades de comunicación, aprendizaje, entretenimiento o de convivencia social.

La programación móvil ha abierto la posibilidad de poner en juego la creatividad, el ingenio, el conocimiento y sobre todo el deseo de satisfacer los requerimientos sociales de esta nueva era, de muchos desarrolladores de software de incontables equipos móviles que ofertan el mercado tecnológico existente.

El desarrollo de aplicaciones móviles se puede realizar mediante diferentes plataformas, que presentan un propio sistema de codificación y de compilación, por lo que el potencial que presenta cada elemento en desarrollo es diferente.

Para poder desarrollar un elemento de aplicación para cualquier dispositivo móvil lo principal a conocer es el medio en que funcionará, es decir el sistema operativo o plataforma final en que se ejecutará y la forma en que lo hará.

Dependiendo del sistema operativo o plataforma en la que se quiera desarrollar la aplicación móvil Web, Nativa o Híbrida, se emplean las herramientas Xcode, Android Studio, Eclipse + SDK, Visual Studio, o cualquier otra; como también de los lenguajes de programación como Objective-C, Swift, Java, C# o XAML, etc.

A continuación se exponen el desarrollo para los tres principales sistemas operativos de dispositivos móviles, como Android, iOS y Window Phone:

2.2.1. Desarrollo para Android

- **Lenguaje de Programación Java**

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems; como cualquier lenguaje tiene su propia estructura, reglas de sintaxis y paradigmas de programación; mismo que se basa en el concepto de programación orientada a objetos (OOP) que las funciones de lenguaje soportan.

```
Ventana.java
1 package ventanas;
2
3 import java.awt.Graphics; // importar la clase Graphics
4 import javax.swing.JFrame;
5
6 public class Ventana extends JFrame {
7
8     // constructora
9     public Ventana() {
10         setTitle("Gráfica"); // título
11         setSize(600,250); // ventana de 600x250
12         setDefaultCloseOperation(EXIT_ON_CLOSE);
13     }
14
15     // función que dibuja la gráfica
16     public void paint(Graphics g) {
17         // para cada coordenada x ...
18         for (int x = 0 ; x < 600 ; x++)
19             // dibujamos una línea entre (x,f(x)) y (x+1,f(x+1))
20             g.drawLine(x, (int)f(x), x + 1, (int)f(x + 1));
21     }
22
23     // función a dibujar
24     double f(double x) {
25         // en Java "cos" se escribe "Math.cos" y "sen" "Math.sin"
26         return 250-(Math.cos(x/3) + Math.sin(x/5) +3) * 40 ;
27     }
28 }
29
30
```

Ilustración 7 Ejemplo de una clase Java

Fuente: informaticaredes-mex.tl

Java es un lenguaje derivado del lenguaje C por lo que sus reglas de sintaxis se parecen mucho a C; por ejemplo, los bloques de códigos se modularizan en métodos y se delimitan con llaves ({y}), y las variables se declaran antes que se usen.

Estructuralmente el lenguaje Java comienza por paquetes, un paquete es un mecanismo de espacio de nombres de lenguaje Java; dentro de estos se encuentran las clases, y dentro las clases se encuentran métodos, variables, constantes, entre otros.

Características de Java

- ✓ *“Es un lenguaje compilado que se interpreta con una máquina virtual.*
- ✓ *La máquina virtual interpreta el código Java (Bytecode) y hace que los programas Java se interpreten en cualquier dispositivo.*
- ✓ *En muchos lenguajes como C o C++ se desarrolla el programa que una vez compilado es ejecutado por el sistema operativo.*
- ✓ *En Java el código fuente se escriben con archivos en extensión.java. El compilador traduce el código fuente a un archivo. class que no es ejecutado por el sistema operativo. El archivo. Class es interpretado por la máquina virtual Java.*
- ✓ *Existen compiladores de Java que traducen a código ejecutable. Exe a partir del Bytecode. Esto incrementa la eficiencia pero el código no es portable.*
- ✓ *Los programas Java son independientes de la plataforma y producen el mismo resultado”.* (Universidad de Nebrija, 2012)

- **Herramientas de desarrollo para Android**

- a) **Android Studio**

Android Studio es un nuevo entorno de desarrollo integrado para el sistema operativo Android diseñado por Google, diseñado para ofrecer nuevas herramientas para el desarrollo de aplicaciones y alternativa al entorno eclipse, hasta ahora el IDE más utilizado.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”



Ilustración 8 Ventana de Android Studio

Fuente: elandroid libre.com

Permite ver los cambios visuales que se realiza en una aplicación en tiempo real, pudiendo a demás ver como se visualiza en diferentes dispositivos Android con distintas configuraciones y resoluciones de forma simultánea.

Android Studio también se caracteriza son sus nuevas herramientas de empaquetado y etiquetado de código para organizaciones al implementar grandes cantidades de código, sirviéndose además de un sistema drag&drop para mover los componentes a través de la interfaz de usuario.

Cuenta además este nuevo entorno con Google Cloud Messaging, una funcionalidad con la que se puede enviar datos desde el servidor a terminales Android a través de la nube, siendo una forma de enviar notificaciones Push a las aplicaciones móviles.

Ayuda también en la localización de las aplicaciones, dando una forma más visual de seguir programando y controlar el flujo de aplicación.

En conclusión Android Studio ofrece:

- ✓ Un entorno de desarrollo claro y robusto.
- ✓ Facilidad para testear el funcionamiento en otros tipos de dispositivos.
- ✓ Asistentes y plantillas para los elementos comunes de programación en Android.
- ✓ Un completo editor con muchas herramientas extra para agilizar el desarrollo de las aplicaciones.

- ✓ Para instalar Android Studio es necesario disponer del Software Kit (SDK) de Android así como Java Developer Kit (JDK), ya incluidas en el paquete de descarga.

b) Eclipse + SDK

Eclipse es un completo entorno de desarrollo integrado para Java y más; es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java.

Es un entorno de desarrollo integrado (IDE)¹⁵, en el que se encuentra todas las herramientas y funciones necesarias para los diferentes proyectos de programación de aplicaciones móviles, cuenta con una interfaz que lo hace fácil y agradable de usar.



Ilustración 9 Applications programs Eclipse

Fuente: eclipse.org

Cuenta con un editor de texto donde se puede ver el contenido del archivo en el que se está trabajando, una lista de tareas, y otros módulos similares.

Las características del programa se pueden ampliar y mejorar mediante el uso de plug-ins¹⁶

¹⁵ **IDE** es un entorno de desarrollo integrado

¹⁶ **Plug-in** es una aplicación que complementa o mejora otra aplicación

2.2.2. Desarrollo para iOS

- **Lenguajes de programación**

- a. Objective-C**

“Es un lenguaje orientado a objetos que es tanto imperativo como declarativo, se caracteriza por la forma de manejar la información; se basa en los conceptos: clase, objeto, herencia” (Castro & Cucker).

Una clase corresponde a un tipo de datos, el término clase es una abreviatura de clase de objetos o conjunto al que corresponde según sea el tipo. Los objetos son dinámicos, es decir pueden ser creados y eliminados en el tiempo de ejecución del programa; los objetos de una misma clase mantienen datos y soportan operaciones similares.



Ilustración 10 Objective-C tercer lenguaje más utilizado

Fuente: applediario.com

La herencia es una facilidad del lenguaje para definir nuevas clases de objetos que son extensión de una clase previamente definida; la nueva clase heredará las propiedades de la clase que se está extendiendo.

El lenguaje orientado a objetos más conocido es Smalltalk que apareció en 1976, es un lenguaje de tipo imperativo donde se utilizó por primera vez el vocabulario que se ha hecho característico. Los términos principales de este vocabulario son, además de los ya conocidos de clase y objeto, los de mensaje y método; la manera de realizar

una tarea es mediante mensajes que son enviados a los objetos, un mensaje es como la llamada a un procedimiento: se indica el objeto a quien va dirigido, el nombre de la operación a realizar y los parámetros necesarios para llevarla a cabo, el método sería la implementación del procedimiento, es decir, cómo realizar la operación.

Un lenguaje que se ha añadido a los llamados orientados a objetos es la extensión de C aparecida en 1986, C++; aunque es diferente a Smalltalk. En C++ la relación de extensión es base/clase derivada, en cambio debido a que este lenguaje es una extensión de un lenguaje imperativo tradicional como es c, los mensajes y los métodos son exactamente las llamadas a procedimiento y las definiciones de procedimientos usuales en C.

“Objective-C es un lenguaje de programación compatible hacia atrás, haciendo que mucha de la sintaxis y características de C estén presentes como por ejemplo

- ✓ *Sentencias de control de flujo (if, for, while)*
- ✓ *Tipos de datos fundamentales, estructuras y punteros.*
- ✓ *Conversiones implícitas y explícitas entre tipos.*
- ✓ *El ámbito de las variables: Global, estáticos o locales.*
- ✓ *Las funciones y su sintaxis.*
- ✓ *Los directivos del preprocesador, (añadiendo objective-V las suyas, así como las llamadas directivos del compilador.” (UDE, 2013)*

b. Swift

Swift es un nuevo lenguaje de programación que ha presentado Apple para los desarrolladores de sus aplicaciones en iOS y OSX, más sencillo y rápido para crear un código robusto, con características similares al lenguaje C o también al Objective-C.

Una de las grandes fortalezas de este lenguaje es su flexibilidad frente a su sintaxis, recoge las mejores características de C y de Objective-C, como tipos, control de flujo u operadores que incluye algunas características enfocadas a la programación, orientada a objetos como las clases y los protocolos así como otras características para hacerle más expresivo:

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

- ✓ Cierres unificados con punteros de función.
- ✓ Múltiples valores de retorno.
- ✓ Interacción rápida y concisa sobre un rango o una colección.
- ✓ Estructuras que soportan, métodos, extensiones y protocolos.
- ✓ Patrones de programación con mapas o filtros.

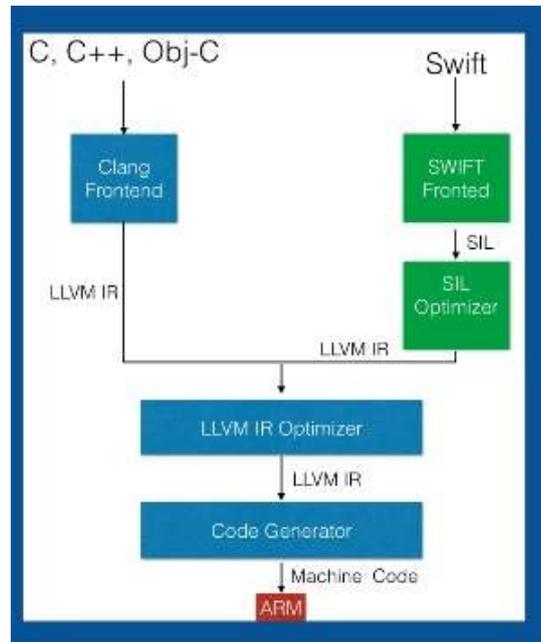


Ilustración 11 Interoperabilidad a nivel de lenguaje

Fuente: es.slideshare.net

- **Herramienta Xcode para desarrollo iOS**

Xcode es el entorno de desarrollo integrado (IDE) creado por Apple que trabaja de manera conjunta con Interface Builder¹⁷ como la herramienta gráfica para la creación de interfaces de usuarios, y que se puede descargar de manera gratuita desde Mac App Store.

Los lenguajes que se pueden utilizar con Xcode son:

- ✓ Compiladores del proyecto GNU (GCC)
- ✓ C, C++, Objective-C, Objective-C++
- ✓ Java y Apples Script
- ✓ Cocoa, Carbón.

¹⁷ **Interface Builder:** es una herramienta gráfica para crear interfaces de usuarios.

Xcode ofrece a los desarrolladores todo lo necesario para crear aplicaciones Mac, iPhone y Pad; la última versión cuenta con un diseño de interfaz de usuario que unifica la codificación, pruebas, y depuración dentro de una única ventana.

2.2.3. Desarrollo para Windows Phone

- **Lenguajes de Programación**

- a. **C#**

- “

- ✓ *C# es un lenguaje de programación que se deriva de los lenguajes C y C++.*
 - ✓ *Necesita NET Framework como entorno de trabajo para iniciar a programar.*
 - ✓ *Emplea SharpDevelop como entorno de programación.*
 - ✓ *Forma parte de la plataforma.NET.*
 - ✓ *Es un interfaz de programación de aplicaciones.*
 - ✓ *Es un lenguaje independiente creado originalmente para producir programas sobre la plataforma.NET.*
 - ✓ *Su código se puede tratar íntegramente como un objeto.*
 - ✓ *Su sintaxis es muy similar a la de JAVA.*
 - ✓ *Es un lenguaje de programación orientado a objetos y a componentes.*
 - ✓ *Armoniza la productividad de Visual Basic con el poder y la flexibilidad de C++.*
 - ✓ *Ahorra tiempo en la programación por contar con abasto completo de librería bien diseñada y estructurada.” (La revista informática, 2009)*

- b. **XAML**

XAML es un lenguaje de programación que permite definir la interfaz del usuario, está basado en XML y es un lenguaje declarativo, pensado para describir la interfaz gráfica de una aplicación de forma textual y ordenada, sus elementos suelen disponer de atributos, lo que permite definir varias opciones.

Ofrece un método basado en herramientas muy sencillo para describir interfaces de usuario, y de este modo permite una mejor colaboración entre desarrolladores y diseñadores; XAML es el lenguaje que subyace a la presentación visual de una

aplicación desarrollada en Microsoft Expression Blend, al igual que HTML es el lenguaje que subyace a la presentación visual de una página web, es un lenguaje que forma parte del WPF. (J.Castellanos, 2012)

- **Herramienta Visual Studio para Windows Phone**

Visual Studio constituye un conjunto de herramientas y otras tecnologías de desarrollo de software basado en componentes para crear aplicaciones eficaces y de alto rendimiento.

Es un entorno de desarrollo integrado para sistemas operativos Windows, soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic, NET, otros; al igual que entornos de desarrollo web como ASP NET MVC, Django. Permite a los desarrolladores crear aplicaciones, sitios, y aplicaciones web, así como también servicios web en cualquier entorno que soporte la plataforma .NET.

Visual Studio incluye plantillas de proyectos, un editor de códigos, un diseñador visual y un cuadro de herramientas de controles, incluye también características de prueba integradas, que permiten al programador realizar cualquier cosa que necesite para desarrollar una aplicación. Cabe indicar que esta herramienta ha sido innovada de acuerdo a los requerimientos de programación, de ahí que se puede descargar libremente Visual Studio una versión para estudiantes.

2.3. TIPO DE APLICACIONES MÓVILES

2.3.1. Aplicaciones Web

Según el diccionario de informática y tecnología, Alegsa Leandro dice *“una aplicación web es cualquier aplicación que es accedida vía web por una red como internet o intranet. Una de las ventajas de las aplicaciones web cargadas desde internet u otra es la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software en, potencialmente, miles de clientes, también la posibilidad de ser ejecutadas en múltiples plataformas”* (Leandro, 2013).

Una aplicación Web es un sitio que contiene páginas estáticas y dinámicas con contenidos sin determinar, parcialmente o en su totalidad; la primera se caracteriza por no cambiar cuando el usuario lo solicita, en este caso el servidor Web envía la

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

página al navegador web solicitante sin modificarla, por el contrario, el servidor modifica las páginas web dinámicas antes de enviarlas al navegador solicitante.

Una aplicación Web tiene varios usos, tanto para los visitantes como para los ingenieros desarrolladores; entre las que se citan las siguientes:

- ✓ “Permitir a los usuarios localizar información de forma rápida y sencilla en un sitio web en el que se almacena gran cantidad de contenido.
- ✓ Recoger, guardar y analizar datos suministrados por los visitantes de los sitios.
- ✓ Actualizar sitios web cuyo contenido cambia constantemente” (Adobe, 2014).



Ilustración 12 Aplicaciones Web

Fuente: miguelangelacera.com

Entre las características principales de la Aplicaciones Web se tiene:

- ✓ “El usuario puede acceder fácilmente a estas aplicaciones empleando un navegador web (cliente) o similar.
- ✓ Si es por internet el usuario puede entrar desde cualquier lugar del mundo donde tenga un acceso a internet.
- ✓ Pueden existir miles de usuarios pero una única aplicación instalada en un servidor, por lo tanto se puede actualizar y mantener una única aplicación y todos los usuarios verán los resultados inmediatamente.

- ✓ *Emplean tecnología como Java, JavaFx, JavaScript, DHTML, Flash, Ajax, otros, que dan gran potencia a la interfaz de usuarios.*
- ✓ *Emplean tecnologías que permiten una gran portabilidad entre diferentes plataformas. Por ejemplo una aplicación web flash podría ejecutarse en un dispositivo móvil, en una computadora con Windows, Linux u otro sistema, en una consola de video juegos, etc.” (Leandro, 2013)*

2.3.2. Aplicaciones Nativas

“Una aplicación nativa es una aplicación software desarrollada para smartphones diseñadas para explotar al máximo las características del dispositivo móvil, estas aplicaciones están implementadas en el lenguaje nativo del propio terminal: Objective-C para iOS, Java para Android o C# para Windows Phone” (Doncel, 2010).

Entre las ventajas de las aplicaciones nativas se tiene:

- ✓ *“Acceso completo al dispositivo.*
- ✓ *Mejor experiencia del usuario.*
- ✓ *Visibilidad en APP Store.*
- ✓ *Envío de notificaciones o avisos a los usuarios.*
- ✓ *La actualización de las app es constante.*

Inconvenientes:

- ✓ *Diferente habilidades/idiomas/herramientas para cada plataforma de destino.*
- ✓ *Tienden a ser más caras de desarrollar.*
- ✓ *El código del cliente no es reutilizable entre las diferentes plataformas.”*
(LanceTalent, 2014)



Ilustración 13 App nativas vs App Web

Fuente: www.eternalsoftware.com.ar

Las aplicaciones nativas requieren un desarrollo para cada sistema operativo, tienen mayor rendimiento, y su interfaz concuerda con el aspecto del sistema en la mayoría de los casos; mientras que las aplicaciones Web funcionan en todas las plataformas, para lo cual debe solo considerarse la compatibilidad con el navegador; sus características las hacen diferentes pero no posible identificar cuál es la mejor, sin embargo a la hora de desarrollar aplicaciones móviles existen muchas opciones a las cuales recurrir.

2.3.3. Aplicaciones Híbridas

Una aplicación híbrida es una aplicación escrita en el mismo lenguaje de las aplicaciones web que está alojada en un contenedor nativo, en un dispositivo móvil. Es la unión entre una tecnología web y la ejecución nativa, una aplicación móvil que se comporta y se parece a una aplicación nativa.

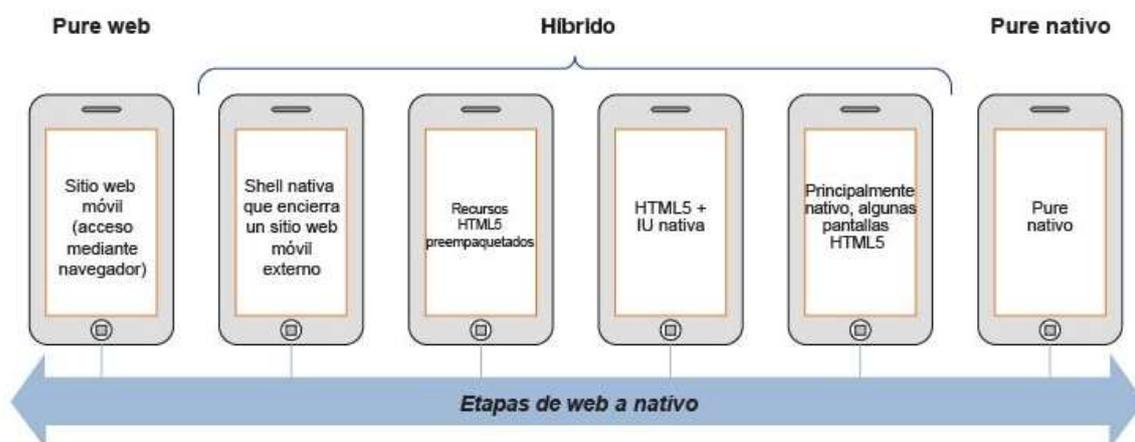


Ilustración 14 Métodos de desarrollo de aplicaciones móviles

Fuente: IBM Worklight Foundation

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

Las aplicaciones híbridas son aquellos desarrollados para dispositivos móviles que permiten:

- ✓ *“Crear un núcleo de código común utilizando tecnologías web.*
- ✓ *Complementarlo de forma opcional con elementos desarrollados con tecnologías nativas de cada plataforma*
- ✓ *Encapsularlo en un contenedor específico para cada plataforma, de manera que la aplicación pueda interactuar con el sistema operativo subyacente, acceder a las capacidades del hardware y puede distribuirse a través de los marketplaces oficiales de cada fabricante como Apple Store, Google Play” (Delgado, 2013)*

Según el mismo autor, las ventajas del desarrollo híbrido frente a un 100% nativo, son:

- ✓ *“Misma experiencia de usuario y mismo rendimiento.*
- ✓ *Menor time-to-market y menores riesgos asociados con el lanzamiento de nuevas aplicaciones.*
- ✓ *Aprovechamiento del conocimiento y los recursos acumulados en el desarrollo con tecnologías web.*
- ✓ *Un núcleo de código común a todas las plataformas”.*

En el campo empresarial es una excelente opción las aplicaciones híbridas, por el mismo hecho de tener lo mejor de las aplicaciones web y nativas.

2.4. HERRAMIENTAS DE DESARROLLO MÓVIL

MULTIPLATAFORMA

2.4.1. IBM Worklight

IBM Worklight está diseñado para proporcionar una plataforma abierta y completa para crear, ejecutar y gestionar aplicaciones HTML5 híbridas y nativas; permite reducir costos de desarrollo y mantenimiento, mejorar el tiempo de comercialización, así como la seguridad y control de las aplicaciones móviles.

IBM Worklight consta de cinco componentes, según mobile first foundation:

- ✓ *“IBM Worklight Studio.- Está diseñado para proporcionar un entorno completo para el desarrollo avanzado de aplicaciones móviles multiplataforma.*

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”



Ilustración 15 Componentes de IBM Worklight

Fuente: www.ltdigital.com.au

- ✓ **IBM Worklight Server:** es middleware optimizado para móviles que funciona como pasarela entre aplicaciones, sistemas de fondo y servicios basados en la nube.
- ✓ **IBM Worklight Device Runtime Components:** Ofrece interfaces de programación de aplicaciones (API) de cliente de tiempo de ejecución, diseñadas para mejorar la seguridad, el gobierno y la usabilidad.
- ✓ **IBM Worklight Application Center:** Le permite configurar una tienda de aplicaciones empresariales que gestiona la distribución de aplicaciones móviles ya listas para su producción.
- ✓ **IBM Worklight Console:** Es una interfaz gráfica de usuario (GUI) para el servidor, adaptadores, aplicaciones y servicios de envío, que le ayudan a gestionar, supervisar e instrumentar aplicaciones móviles.” (IBM Worklight, 2014)

2.4.2. PhoneGap

“PhoneGap es un framework para el desarrollo de aplicaciones nativas de sistemas operativos móviles, haciendo uso de tecnologías web como HTML5, CSS3 y JavaScript, es posible desarrollar aplicaciones para los siguientes sistemas operativos como Android, iOS, Windows Phone, BlackBerry OS, Web OS, Symbian y Bada; inicialmente fue desarrollado por Nitobi bajo licencia de software libre,

pero para octubre de 2011, Adobe anunció oficialmente la adquisición de Nitobi, pasando, así PhoneGap al control del gigante del software, gran interesado en la evolución de HTML5. Esto generó una gran incertidumbre entre los desarrolladores, pues el framework podía pasar a ser una tecnología propietaria pero en Apache, conservando de esta forma la integridad libre de PhoneGap”. (Galeano, 2012)

“Dentro de las principales características de PhoneGap nos encontramos una serie de Apis para controlar los diferentes recursos del dispositivo como lo son: cámara, acelerómetro, GPS, notificaciones, almacenamiento, sistema de ficheros, compas, media”. (Pimienta, 2014)

2.5. METODOLOGÍA SCRUM

De entre la variedad de metodologías ágiles para desarrolladores de aplicaciones móviles y demás programas informáticos se emplea SCRUM, que se caracteriza por conseguir resultados óptimos en tiempos cortos, con el monitoreo constante y la intervención oportuna mediante el trabajo en equipo; donde juegan roles importantes y específicos tanto el cliente propietario del sistema, la persona coordinadora del grupo y lógicamente los desarrolladores programadores.

Lo interesante de aplicar la metodología SCRUM consiste el trabajo colaborativo que se hace para lograr la satisfacción del cliente; siendo las interacciones personales lo que alimenta a la empresa desarrolladora y la fortalece, garantizando el éxito del proyecto en proceso.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

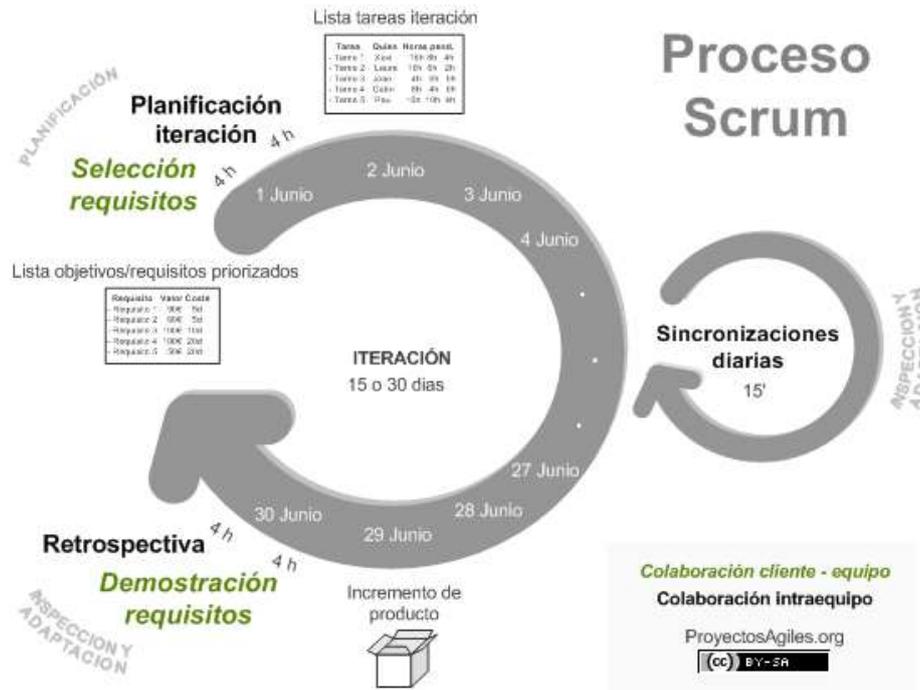


Ilustración 16 Diagrama del Proceso SCRUM

Fuente: www.lafabricadeltiempo.es

El proceso de una metodología SCRUM es muy dinámico, interactivo, incremental, retrospectivo y funcional; empieza con un primer contacto e interacción con el cliente (Product Owner), donde el coordinador del equipo de desarrolladores (SCRUM Master) está listo a recibir todos sus requerimientos respecto del proyecto informático, mismos que se priorizan conjuntamente con el equipo de desarrolladores, convirtiéndose en los objetivos del plan a ser cumplidos en un tiempo corto establecido, con la única intención de prestar servicios de calidad que satisfagan al cliente.

Una fase importante dentro del proceso de SCRUM es la planificación de los sprints o interacciones diarias del equipo, donde a cada desarrollador parte del equipo le queda claro qué actividades deben cumplirse y el tiempo establecido para el efecto; sin embargo cada día por un tiempo máximo de 15 minutos se reúne el equipo para hacer un balance, aportes o reajustes necesarios y oportunos.

El SCRUM Master cumple un rol muy importante dentro del equipo, generando un ambiente de trabajo saludable y motivante, de tal modo que cada interacción sea productiva que conlleve a maximizar la productividad de la empresa; inclusive cuando ya está finalizado el proyecto, él es quien junto con su equipo de trabajo pone a consideración la demostración respectiva al cliente.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

La fase retrospectiva permite evaluar todo el proceso seguido; tanto las interacciones, incrementos, adaptaciones o correcciones realizadas, para re planificar el proyecto si el caso lo amerita.

La metodología SCRUM se podría resumir en tres fases importantes: Planificación de interacciones, sincronizaciones diarias y fase retrospectiva, en las cuales se cumplen actividades como: Planificación de las interacciones, ejecución de las interacciones, reuniones diarias sincronizadas, demostración de funcionamiento del desarrollo, revisión del proceso inicio-fin (retrospección) y re planificación del proyecto (solo si es necesario).

Se ha escogido la metodología SCRUM para el desarrollo de la aplicación móvil con IBM Worklight porque como se ha indicado es una metodología ágil que permite en un periodo de tiempo corto de cuatro a cinco semanas lograr los objetivos propuestos dentro del proyecto de desarrollo del software aplicativo, de modo que el cliente que en este caso es el Instituto Tecnológico Superior “José Chiriboga Grijalva”. (ITCA) quede satisfecho con el trabajo final desarrollado por un estudiante de la prestigiosa Universidad Técnica del Norte de la cual soy parte, con la asesoría profesional del Ingeniero Diego Trejo tutor de mi trabajo de grado.

MARCO INVESTIGATIVO

3. IBM WORKLIGHT STUDIO

3.1. IBM WORKLIGHT FOUNDATION

IBM es una empresa que desde su creación ha sido uno de las pioneras en crear tecnología tanto en hardware como software.

IBM en su afán de crear más tecnología decidió crear dentro empresa equipos de trabajo que logren mejorar el hardware y software para desarrollo, otra opción que esta empresa suele optar es comprar ideas por parte de otras empresas, y seguir investigando y mejorando la calidad de software y hardware de la misma.

Este es el caso de IBM Worklight Foundation, que fue creado por Worklight, una empresa que se dedicaba a la investigación y software de desarrollo móvil, el objetivo de IBM con esta adquisición es ampliar sus negocios y los negocios de todo el mundo al mundo móvil.

“IBM Worklight Foundation proporciona una plataforma de aplicación móvil avanzada, abierta y completa para teléfonos inteligentes y tabletas. Ayuda a organizaciones de todo tipo a desarrollar, probar, conectar, ejecutar y gestionar de manera eficiente aplicaciones móviles. Mediante tecnologías y herramientas basadas en estándares, IBM Worklight Foundation ofrece una plataforma integrada que incluye un entorno de desarrollo completo, middleware de tiempo de ejecución optimizados para móviles, una tienda de aplicaciones empresariales privada y una consola de gestión y análisis integrada; todo esto admitido por varios mecanismos de seguridad.” (IBM Worklight Foundation, 2014)

IBM Worklight Foundation ofrece componentes que ayudan a cumplir todo lo que antes fue mencionado, estos componentes son:

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

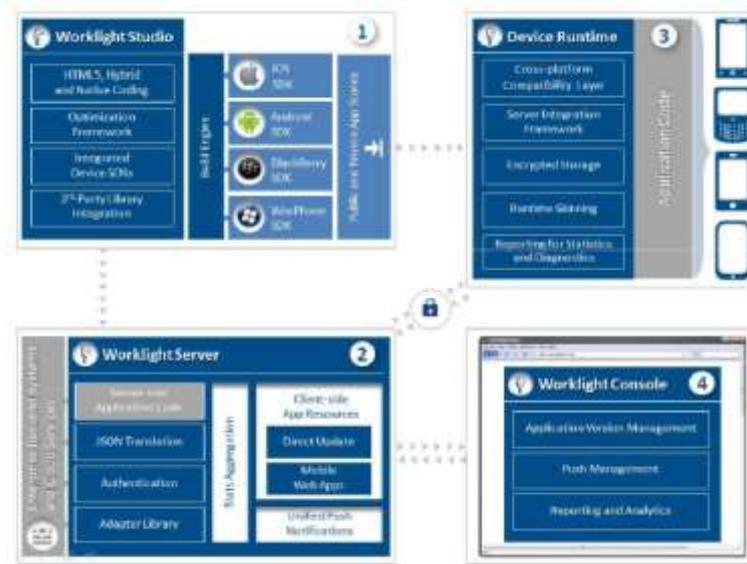


Ilustración 17 Componentes de Worklight

Fuente: IBM Worklight Foundation

- **Worklight Studio**

En una plataforma de desarrollo móvil, que pretende la portabilidad de código ya que una característica importante que posee es que puede desarrollar a multiplataforma, lo que ayuda a los programadores a reutilizar el mismo código para diferentes lenguajes de programación móvil.

Además posee herramientas propias de esta herramienta para lograr realizar un desarrollo multiplataforma.

- **Worklight Server**

Este servidor es un contenedor en tiempo de ejecución de las aplicaciones móviles que son desarrolladas en Worklight Studio, que actúa como un contenedor de paquetes de aplicación.

- **Componentes Client-side runtime**

Esta API de tiempo de ejecución de cliente son varias bibliotecas integradas en el código de la aplicación móvil. IBM Worklight Foundation utiliza la infraestructura de desarrollo de Apache Cordova, que ofrece un puente constante entre tecnologías de web estándar y las funciones nativas de las diferentes plataformas móviles.

- **Worklight Console**

Se utiliza para controlar y gestionar las aplicaciones móviles. Recopila datos y estadísticas de uso de las aplicaciones para que se analicen con los sistemas Business Intelligence de la organización.

- **Application Center**

Es un repositorio donde se puede compartir nuevas aplicaciones móviles en la organización, también se puede usar para compartir aplicaciones en desarrollo entre los miembros del equipo de trabajo que estén implicados en el desarrollo de las mismas.

- **IBM Mobile Application Platform Pattern**

Se puede desplegar Worklight Server en IBM PureApplication System o IBM SmartCloud Orchestrator. Dando la opción a las organizaciones a aprovechar las tecnologías en la nube.

3.1.1. Novedades de la Herramienta

En los primeros días del año 2015, IBM Worklight Foundation cambió de nombre para posicionarse mejor en el mercado, ya que muchas personas desconocían su uso mediante el nombre.

Por lo que se cambió el nombre a IBM MobileFirst Platform Foundation, que posee las mismas características, herramientas y componentes que la anterior, pero en diferente nombre.

Se han añadido dos herramientas más, pero los cambios se dieron con lo que respecta a la interacción con usuarios de los otros componentes de esta plataforma.

En esta investigación se centrará al estudio de uno de los componentes de esta fundación que es IBM Worklight Studio.

3.2. CARACTERÍSTICAS DE IBM WORKLIGHT STUDIO

Antes de describir las características de esta herramienta es necesario recalcar que IBM Worklight Studio es un plug-in¹⁸ de Eclipse¹⁹ que contiene una versión embebida de IBM Worklight Server.

“IBM Worklight Studio es un IDE basado en Eclipse que permite a los desarrolladores realizar prácticamente todas las tareas de codificación e integración que requieren para desarrollar aplicaciones enriquecidas dirigidas a los empleados y clientes.

IBM Worklight Studio enriquece las herramientas familiares de Eclipse con una amplia variedad de características de nivel empresarial ofrecidas por los IBM Worklight Plug-ins, permitiendo que IBM Worklight Studio dinamice el desarrollo de aplicaciones y facilite la conectividad empresarial.” (IBM Corporation, 2013)

3.2.1. Soporte Multiplataforma

“IBM Worklight Studio permite el desarrollo de aplicaciones móviles nativas, híbridas y Web enriquecidas en tablets y smartphones en iOS, Android, Windows Phone, BlackBerry y tabletas y smartphones con Windows 8.” (IBM Corporation, 2013)

IBM Worklight Studio posee un framework²⁰ de optimización, que hace que los desarrolladores al estar programando para un ambiente determinado no pierdan más tiempo al querer realizar lo mismo para otro ambiente, sino compartir la mayoría de código y dividirlos en carpetas separadas para la implementación de la misma pero en otro ambiente. La lógica de programación permanece en todos los ambientes permitiendo agregar funcionalidades o borrarlas según los objetivos del desarrollador.

“La lógica de la aplicación permanece consistente entre los distintos ambientes, mientras que la interfaz de usuario (IU) se comporta de forma nativa y cumple con las expectativas del usuario y las pautas de funcionalidad y diseño únicas del dispositivo.” (IBM Corporation, 2013)

¹⁸ **Plug-in:** es un complemento de otra aplicación cuyo objetivo es aportar funciones nuevas a la aplicación.

¹⁹ **Eclipse:** es un IDE de programación para lenguajes como java, c++, permite realizar proyectos para escritorio, web y móvil. Es libre de paga.

²⁰ **Framework:** es una estructura conceptual y tecnológica de soporte definido, es de gran ayuda en el desarrollo ágil de un proyecto ya que dentro de ella posee herramientas ya definidas para un objetivo de desarrollo.

Además los desarrolladores pueden acceder directamente a las API's²¹ de los dispositivos modernos lo que permite que se puedan integrar más fácilmente con frameworks, herramientas y bibliotecas de terceros. Lo que al final se obtiene es aplicaciones móviles construidas para las necesidades específicas.

3.2.2. Codificación Híbrida

El significado de la palabra híbrida dentro del ambiente de desarrollo se suele confundir con el hecho de ser multiplataforma, o que dentro del IDE se posean herramientas, API's o Aplicaciones pre diseñadas que ayuden a mejorar la visualización de las aplicaciones. Pero estas no son híbridas, ya que para lograr una verdadera codificación híbrida se tiene que realizar combinaciones entre web y nativo en varias formas.

IBM Worklight Studio es una de las herramientas con las que se puede lograr aplicaciones híbridas. Ya que dentro de la programación de la aplicación permite programar con código nativo puro (Objective-C²², Java²³ o C#²⁴) como también en tecnologías Web Estándar (HTML5²⁵, CSS3²⁶ y JavaScript²⁷) y a su vez se pueden realizar una combinación entre las mismas.

- **Escenarios Híbridos**

“

1. *HTML se puede utilizar para llamar código nativo utilizando el plug-in Cordova (anteriormente conocido como PhoneGap²⁸).*

El código nativo puede ser "sin interfaz de usuario", por ejemplo, lectura de la brújula, o realmente puede mostrar una interfaz de usuario (UI). Este

²¹ **API's:** Application Programming Interface, es el conjunto de funciones y procedimientos, que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

²² **Objective-C:** es un lenguaje de programación orientado a objetos creado como un superconjunto de C.

²³ **Java:** es un lenguaje de programación orientado a objetos y basado en clases, fue diseñado específicamente para tener pocas dependencias de implementación como fuera posible

²⁴ **C#:** es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Su sintaxis deriva de lenguajes como C y C++.

²⁵ **HTML5:** (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML.

²⁶ **CSS 3:** siglas en inglés de cascading style sheets) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML2 (y por extensión en XHTML).

²⁷ **JavaScript:** (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos.

²⁸ **Phonegap:** es un framework para el desarrollo que permite a los programadores desarrollar aplicaciones para dispositivos móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3.

componente de interfaz de usuario se superpone en la parte superior del navegador, para que el usuario vea un componente nativo mezclado con la interfaz de usuario HTML.

- 2. Los desarrolladores pueden disminuir el tamaño del navegador para que el navegador ocupe solo una parte de la pantalla. El resto de la pantalla puede utilizarse para mostrar los componentes nativos de interfaz de usuario.*
- 3. Los usuarios pueden implementar pantallas completas nativamente y pueden cambiar entre pantallas nativas y Web. La transición entre las pantallas puede ser animada, como en las transiciones de pantalla regulares.” (IBM Corporation, 2013)*

Además de todo esto IBM Worklight Studio ofrece dentro del desarrollo herramientas de diseño tipo drag-drop²⁹ tanto de herramientas nativas como tecnologías web, estas se encuentran dentro de una paleta que se puede incrustar directamente al código HTML y muestran la visualización inmediata de la misma.

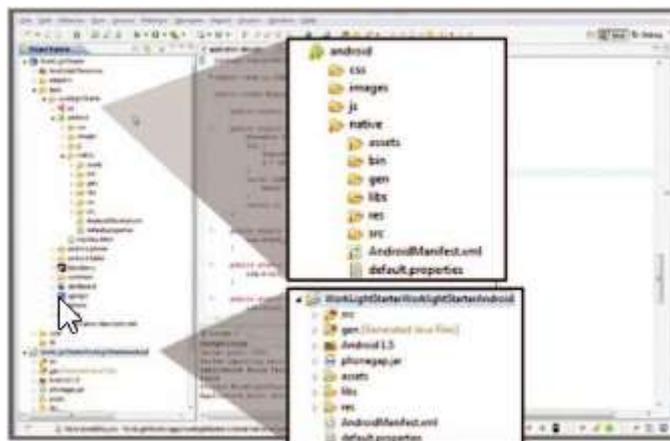


Ilustración 18 Elementos de un Proyecto de Android con IBM Worklight Studio

Fuente: IBM Worklight Foundation

3.2.3. Runtime Skin

La optimización de la ejecución de la aplicación en diferentes dispositivos móviles es posible, gracias a los runtime Skin que son máscaras de diferentes dispositivos que se

²⁹ **Drag-drop:** arrastrar y soltar en español, es una expresión informática que se refiere a la acción de mover con el ratón objetos de una ventana a otra o entre partes de una misma ventana.

empaquetan con los archivos ejecutables de la aplicación. La principal función que posee es ajustarse a los diferentes dispositivos de la misma familia de sistemas operativos.

Los escenarios comunes que posee el Runtime Skin incluyen:

- Diferentes tamaños de pantalla
- Densidad de Pantalla Diferentes
- Diferente método de entrada
- Diferentes Niveles de HTML5



Ilustración 19 Escenarios del Runtime Skin

Fuente: IBM Worklight Foundation

3.2.4. Soporte para HTML5

IBM Worklight Studio aprovecha el estándar de html5 para insertar código HTML puro sin necesidad de utilizar compilación cruzada o transcodificación. Esto evita la limitación de los intérpretes propietarios o de los traductores de código.

Los beneficios que se obtiene son los siguientes:

“

- *Un código HTML más limpio, más legible y coherente.*
- *Acceso a tipos de multimedia enriquecidos (audio y video), disponible previamente solo a través de código nativo.*
- *El uso de componentes de interfaz de usuario avanzados, como recolectores de datos, barras de desplazamiento (sliders) y cuadros de edición que soportan automáticamente elipsis y otros implementados de forma nativa por el navegador.*

- *Uso de estilos Cascading Style Sheets 3 (CSS3) y animación basada en CSS3 para reducir el tamaño de la aplicación y mejorar la capacidad de respuesta de la aplicación.*
- *Canales de distribución de aplicaciones que van más allá de las diferentes tiendas de aplicaciones y sus restricciones limitantes y que requieren de tiempo.*
- *Soporte para servicios de geolocalización.*
- *Capacidades de almacenamiento offline. ”*

3.2.5. Integración del SDK³⁰ de dispositivo nativo

IBM Worklight Studio permite integrarse fácilmente a los SDK de los dispositivos móviles.

De esta manera se aprovecha en gran manera las capacidades nativas de cada SDK, permitiendo que la ejecución y depuración del proyecto se las realice de manera rápida.

“Para dinamizar aún más el proceso de desarrollo iterativo, IBM Worklight Studio mejora las funciones de pre visualización para aplicaciones híbridas iOS y Android. Un simulador en el navegador permite definir el factor de forma del dispositivo de destino, desplegando al mismo tiempo varios dispositivos en la pantalla y simulando las API’s Apache Cordova (un marco de código abierto para hacer puente de llamadas entre componentes nativos y vistas Web)” (IBM Corporation, 2013)

3.2.6. Recuperación estandarizada de datos

“IBM Worklight studio permite a los desarrolladores usar transformaciones XSL y código JavaScript para convertir datos jerárquicos recuperados desde cualquier sistema de back-end³¹ a formato JavaScript Object Notation (JSON), preparando los datos para la distribución y consumo de la aplicación. Los desarrolladores pueden invocar servicios de back-end directamente dentro del studio y pueden recibir resultados sin procesar en Extensible Markup Language (XML), o resultados procesados (después

³⁰ **SDK:** (siglas en inglés de software development kit) es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, etc.

³¹ **Back-end:** En diseño de software el front-end es la parte del software que interactúa con el o los usuarios y el back-end es la parte que procesa la entrada desde el front-end.

de haber convertido a JSON usando transformaciones de Extensible Stylesheet Language [XSL] y JavaScript) en formato JSON.” (IBM Corporation, 2013)

Esto hace que se puedan recopilar datos del lado del servidor con javascript o java, con estas se pueden crear mashups³² de aplicaciones de back-end.

Lo que hace que la red no se haga lenta por todas las peticiones que se realiza y también mejora la capacidad de respuestas de las mismas.

3.2.7. Notificaciones Push unificadas

El desarrollo de la aplicación cuenta con adaptadores de integración, que los desarrolladores pueden aprovechar la arquitectura la Arquitectura Push unificada.

Esto hace que aunque se desarrolle para diferentes dispositivos estos puedan utilizar las misma alertas pre configuradas.

“Worklight hace que todo el proceso de comunicación con los usuarios y dispositivos sea totalmente transparente para el desarrollador” (IBM Corporation, 2013)

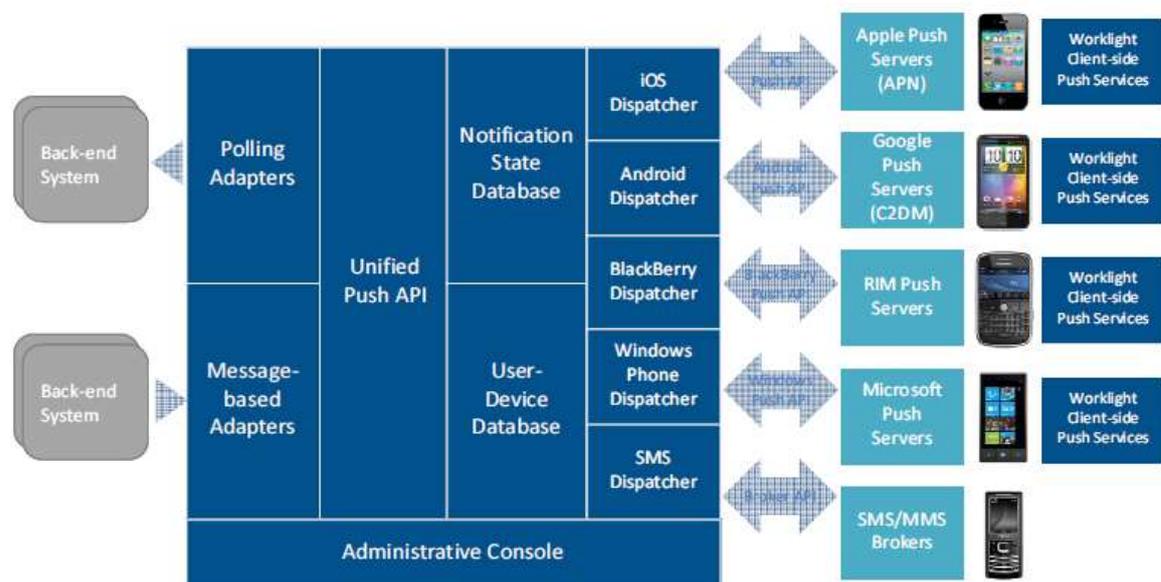


Ilustración 20 Infraestructura de Notificaciones

Fuente: IBM Worklight Foundation

³² **Mashups:** es la integración de una aplicación que es usada desde otra aplicación y con el fin de reutilizar su contenido y funcionalidad en otra fuente, para crear un nuevo servicio simple, visualizado en una única interfaz gráfica.

3.3. COMPONENTES

Los componentes forman parte de IBM Worklight Studio y son esenciales para la creación de proyectos multiplataforma. Estos componentes hacen de IBM Worklight Studio una herramienta que proporciona aplicaciones con escalabilidad.

Los componentes que se hablarán en esta investigación forma parte del desarrollo de un proyecto, y cada una se desarrolla de acuerdo a los objetivos que el programador o desarrollador tenga con respecto a lo que va a desarrollar.

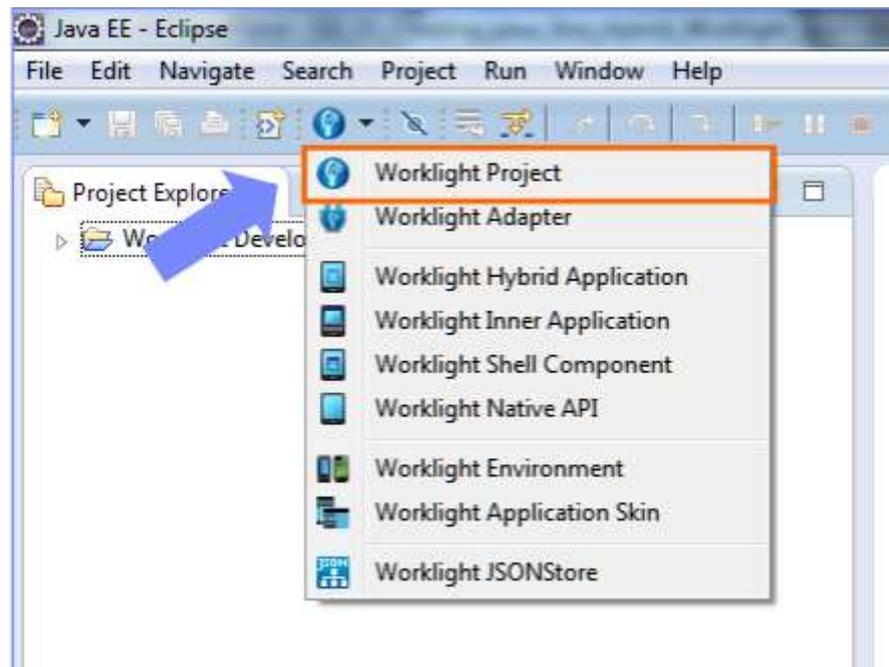


Ilustración 21 Proyecto Worklight

Fuente: IBM Worklight Foundation

3.3.1. Project

Para desarrollar una aplicación móvil sea solamente para un ambiente o más ambientes es necesario crear un proyecto.

El componente Project que ofrece IBM Worklight Studio ofrece la capacidad de crear un proyecto estructurado mediante el ambiente con el que se va a trabajar.

- **Ambientes**

Se puede definir cuatro tipos de ambientes para la creación de un Project

- **Aplicación híbrido:** Una aplicación híbrida puede apuntar a múltiples entornos. Se puede escribir con HTML5, CSS y JavaScript. Se puede acceder a las capacidades del dispositivo mediante el uso de la API de JavaScript de IBM Worklight.
- **Aplicación interna:** Una aplicación interna contiene los HTML, CSS y JavaScript partes que se ejecutan dentro de un componente Shell. Antes de que pueda implementar esta aplicación, debe empaquetarla en un componente de la cubierta para crear una aplicación híbrida completa.
- **Componente Shell:** Un componente Shell proporciona capacidades de encargo nativos y funciones de seguridad que una aplicación interna puede utilizar.
- **Aplicación nativa:** una aplicación nativa se dirige a un entorno específico, y se puede utilizar la API de IBM Worklight para la integración, la seguridad y la gestión de aplicaciones.

Después de definir el tipo de proyecto se procede a crear la aplicación o aplicaciones que se requieran crear, pero la explicación de estas se darán más a continuación.

- **Estructura**

Al crear un proyecto IBM Worklight Studio se crean una estructura de carpetas que ayudan con el desarrollo de la aplicación.

Esta estructura es definida de la siguiente manera:

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

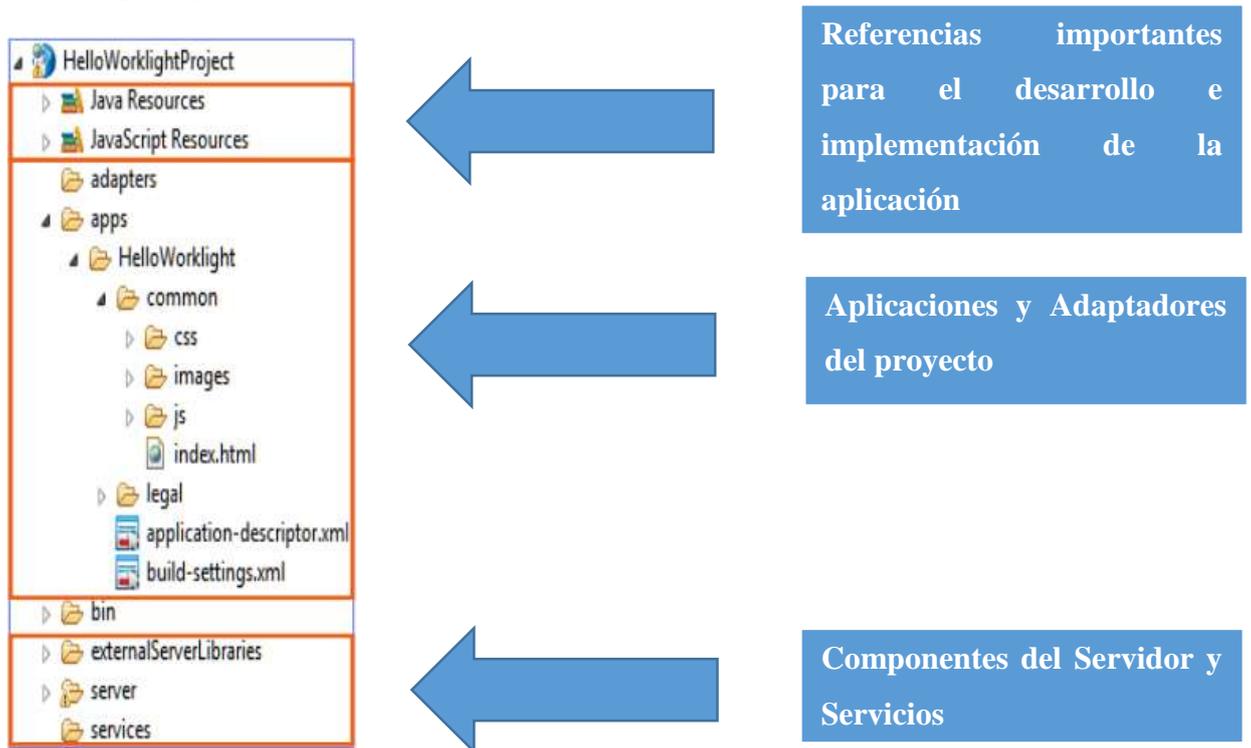


Ilustración 22 Estructura de un proyecto Worklight

Fuente: IBM Worklight Foundation

- **Referencias Importantes**

Dentro de esta se encuentran las librerías que se van a necesitar para el desarrollo de la aplicación móvil, las que se crean por defecto son librerías de Java y JavaScript.

- **Aplicaciones y Adaptadores**

Se crean por defecto la carpeta **common** y **legal**.

Common (común): dentro de ella se almacenan todos los recursos que se comparten entre ambientes. Lo que contiene es un archivo html, carpetas de css, imágenes y js que por defecto viene creado con los métodos de inicialización y comunicación.

Legal: contiene la metadata³³ de la aplicación, opciones de configuración para minificación y concatenación y los bin.

³³ **Metadata:** son datos que describen otros datos

○ Componentes del Servidor

Todo lo referente a configuración de servicios y servidores.

3.3.2. Adapters

Los adaptadores son capas de transporte que IBM Worklight Studio para conectarse a variados servidores.

Los adaptadores se utilizan para recuperar información, para realizar acciones.

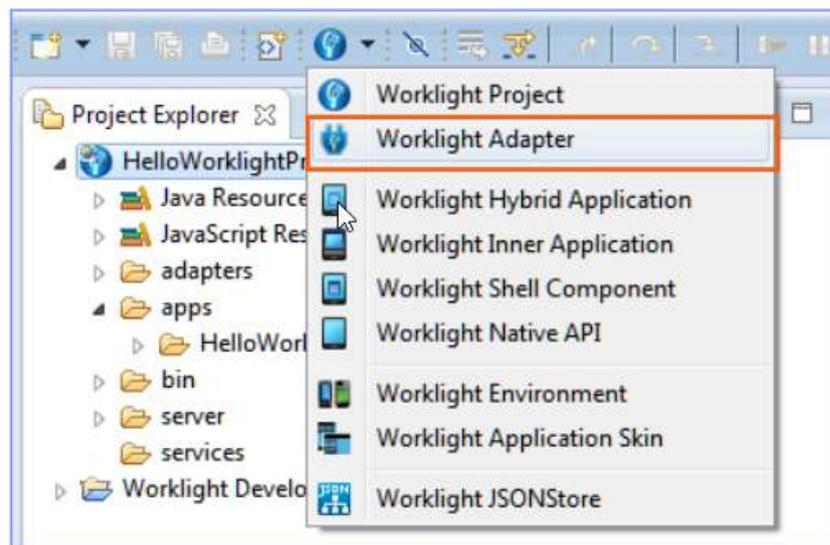


Ilustración 23 Proyecto de Worklight Adapter

Fuente: IBM Worklight Foundation

Cada adaptador que posee IBM Worklight Studio tiene la función de realizar una actividad específica como los siguientes:

- SQL³⁴ Adapter: para conectarse con la base de datos.
- HTTP³⁵ Adapter: soporta protocolos REST³⁶ y SOAP³⁷.
- Cast Iron Adapter: para tener conexión a aplicaciones SOA desarrollado en la plataforma Cast Iron producto de IBM.

³⁴ **SQL:** es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

³⁵ **HTTP:** (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.

³⁶ **REST:** es una técnica de arquitectura de software para sistemas hipermedia distribuidos como la World Wide Web.

³⁷ **SOAP:** (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

- JMS³⁸: para conectarse a la cola de mensajes para la plataforma Java2.
- SAP Netweaver Gateway: para conectarse a la plataforma SAP.

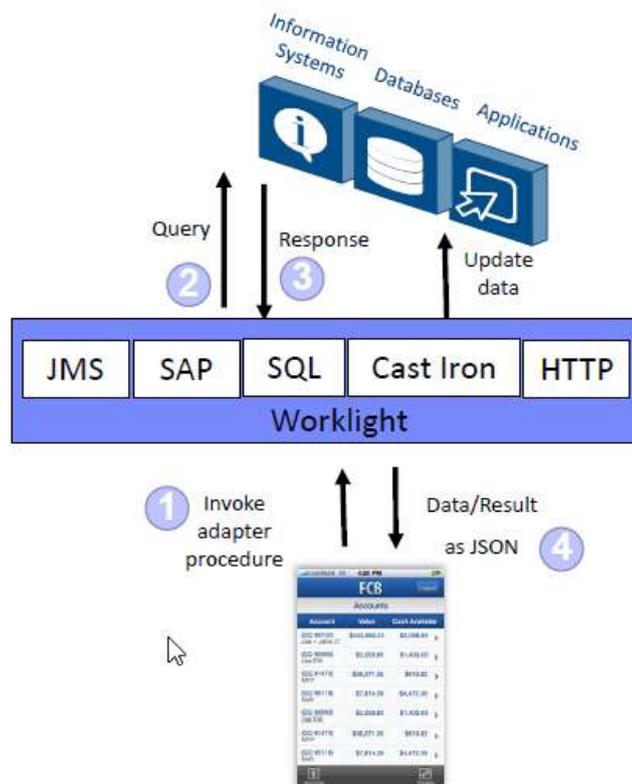


Ilustración 24 Comunicación base de datos con JSON y Worklight

Fuente: IBM Worklight Foundation

Cada adaptador posee la misma estructura, pero lo que los hace diferente es que al llamar a las funciones del archivo JavaScript de cada una ellas se las realiza de manera distinta.

- **BENEFICIOS**

- Soporta la tecnología de integración múltiple y sistemas de información back-end.
- Los adaptadores soportan read-only³⁹ y modos de acceso transaccional para sistemas back-end.
- Usa sintaxis XML simple y fácilmente de configurar con el API de Javascript.

³⁸ **JMS**: Java Message Service, es un estándar de mensajería que permite a los componentes de aplicaciones basados en la plataforma Java2 crear, enviar, recibir y leer mensajes. También hace posible la comunicación confiable de manera síncrona y asíncrona.

³⁹ **Read-only**: es un atributo que se da a algunos elementos de programación en el que solo se permita la lectura.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

- Posee facilidad de autenticación flexible para crear conexiones con sistemas back-end.
- Los adaptadores detectan a los usuarios conectados.
- Mediante el uso del caché, reducen el número de transacciones en sistemas back-end.
- Los datos que se extraen de aplicaciones back-end se exponen de manera uniforme, y de forma independiente del tipo de adaptador.

- **ANATOMÍA**

- Cada adaptador web posee:

Archivo XML: describe las opciones de conectividad y una lista de procedimientos que posee la aplicación.

Archivo Javascript: posee la implementación de procedimientos que están declarados en el XML.

Archivo XSL: posee transformaciones de esquema del archivo XML

- Los datos procesados por los adaptadores son retornados como objetos JSON.
- Para la recuperación de datos se llaman servicios back-end.
- Usa JavaScript del lado del servidor, por lo que se puede procesar un procedimiento antes y después de llamar al servicio.
- Se puede filtrar los datos gracias a un archivo XSLT.
- Se puede utilizar Java en fragmentos de código del adaptador.

- **Estructura**

La estructura de un adaptador es simple ya que es un Archivo XML.

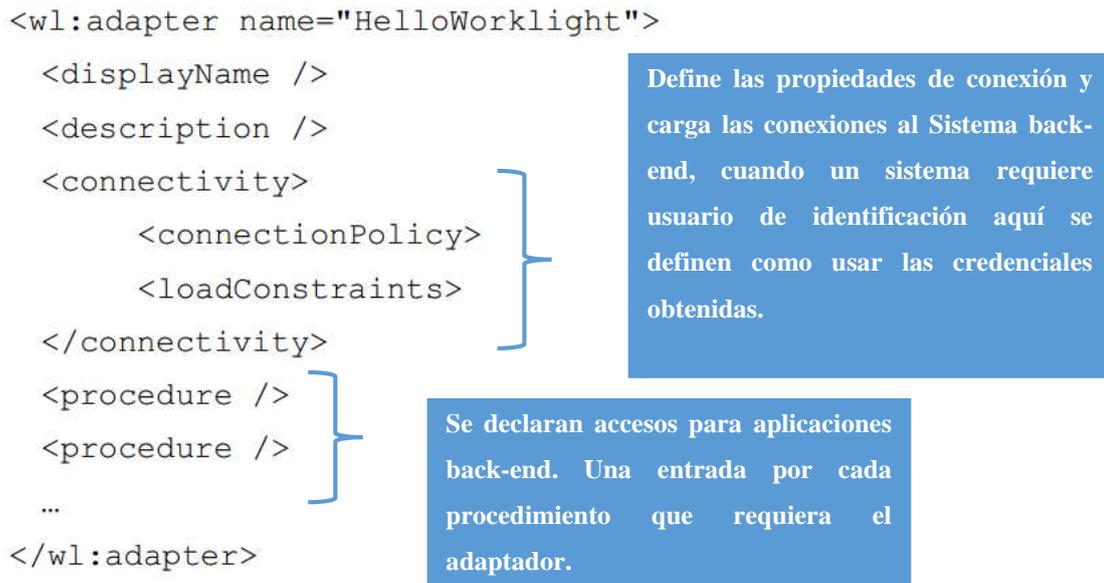


Ilustración 25 Estructura de una adaptador de IBM Worklight

Fuente: IBM Worklight Foundation

Por cada procedimiento declarado en el adaptador (Archivo XML), se posee una función en el archivo JavaScript.

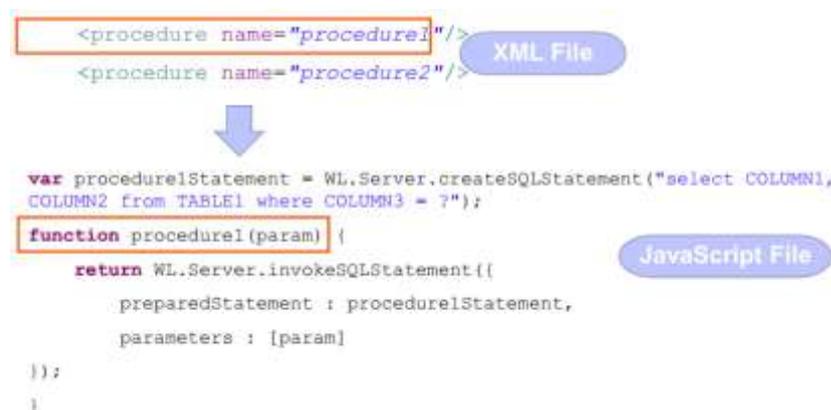


Ilustración 26 Estructura de un adaptador con función Javascript y XML

Fuente: IBM Worklight Foundation

3.3.3. ENVIROMENT

Si bien el ambiente de desarrollo en el que trabaja IBM Worklight Studio es el IDE Eclipse, pero esta es de forma general; dentro de esta herramienta se tienen diferentes ambientes de desarrollo móvil de acuerdo al sistema operativo en el que se pretende desarrollar.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

Al crear un proyecto de IBM Worklight Studio como se había explicado antes se tiene creado un ambiente por defecto, pero en caso de querer realizarlo multiplataforma es necesario llamar a otros ambientes.

Antes de crear otros ambientes hay que crear en las páginas de los desarrolladores de cada ambiente una cuenta, ya que de esa manera permitirá acceder a todos los beneficios y funciones del ambiente que se va a desarrollar.

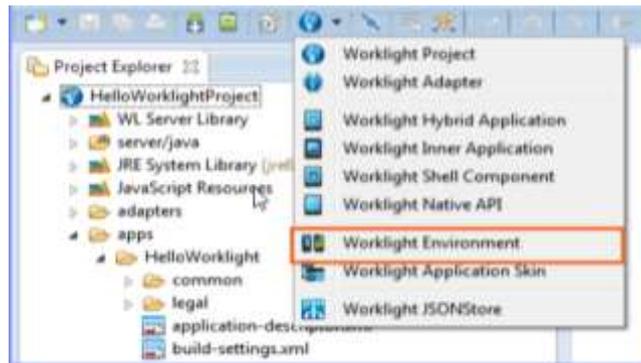


Ilustración 27 Proyecto de Worklight Environment

Fuente: IBM Worklight Foundation

- **Tipos de ambientes**

Como se había mencionado anteriormente hay diversos ambientes para los que se puede programar una aplicación web. IBM Worklight Studio posee los siguientes ambientes para el desarrollo de aplicaciones en general.

- **Mobile:** dirigido al desarrollo de aplicaciones móviles como iPhone⁴⁰, iPad⁴¹, tablets y celulares con Sistemas Operativos Android, BlackBerry y Windows Phone.
- **Desktop:** aplicaciones que sean para Windows 8 y Adobe AIR⁴².
- **Web:** Para aplicaciones web móviles y para navegadores

⁴⁰ **iPhone:** es una línea de teléfonos inteligentes diseñada y comercializada por Apple Inc. Ejecuta el sistema operativo móvil iOS.

⁴¹ **iPad:** es una línea de tabletas diseñadas y comercializadas por Apple Inc.

⁴² **Adobe Air:** es un entorno de ejecución multiplataforma para la creación de aplicaciones RIA (Rich Internet Applications) que utilizan Adobe Flash, Adobe Flex, HTML y AJAX, para usarse como aplicaciones de escritorio.



Ilustración 28 Creación de un Ambiente Worklight

Fuente: IBM Worklight Foundation

3.3.4. SKINS

Para IBM Worklight Studio es muy importante que todas las aplicaciones que se realizan dentro de esta herramienta tengan todas las opciones y herramientas necesarias para que estas tengan un estilo.

El skin⁴³ se los define después de crear el Proyecto, y no es más que una carpeta que posee: imágenes, archivos CSS y JavaScript.

Se puede editar como crear nuevos archivos dentro del Skin, pero la estructura suele mantenerse, por lo que es aconsejable que se creen con diferentes nombres pero se copie la estructura del archivo se da el skin por defecto y editarlo en las partes que se crean convenientes.

⁴³ **Skin:** es una capa de estilo que se le da ese nombre por defecto por ser la piel de una aplicación.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

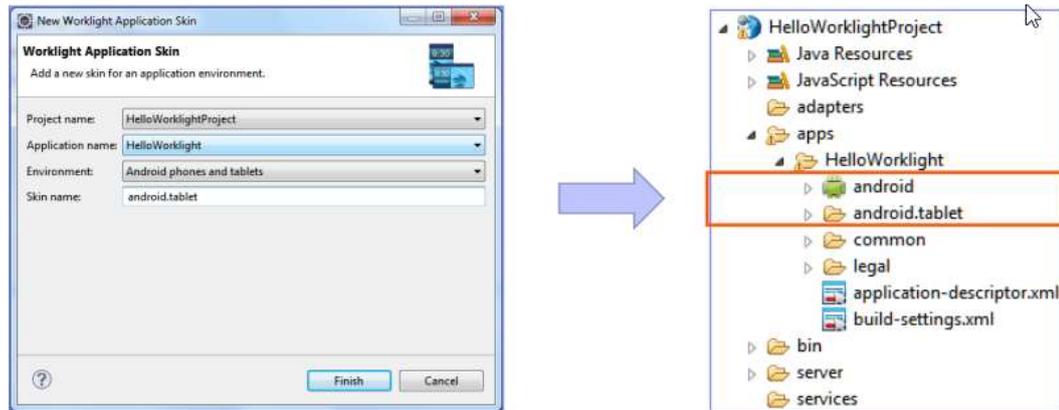


Ilustración 29 Estructura generada de un Worklight Application Skin para Android

Fuente: IBM Worklight Foundation

- **Ejecución en tiempo real**

Muchos desarrolladores eligen el tiempo de ejecución de los skin, estos se lo puede realizar mediante los archivos JavaScript que vienen por defecto en la creación del skin.

Se procede a configurar cada uno de ellos para la ejecución de los skin.

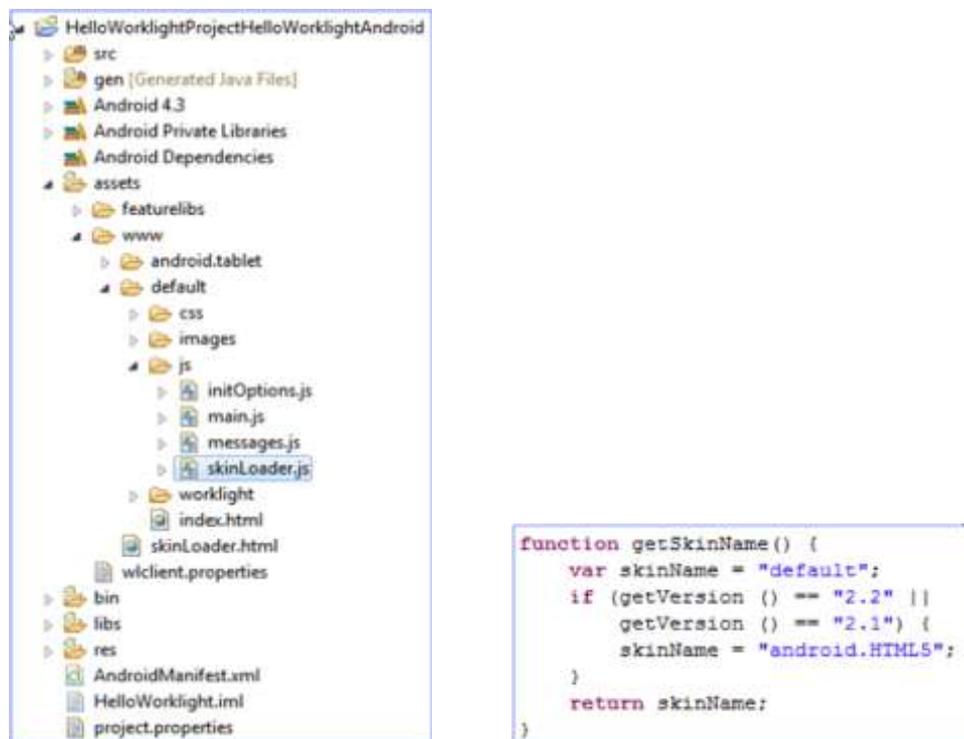


Ilustración 30 Método skinLoader

Fuente: IBM Worklight Foundation

- **JSONSTORE**

Es un API opcional de lado cliente que permite una mejor indexación de datos para mejorar la búsqueda y también ofrece soporte para múltiples usuarios.

“JSONStore es un sistema ligero, orientado a documentos de almacenamiento que se incluye como una característica de la IBM Worklight, permite el almacenamiento persistente de los documentos JSON. Los documentos de una aplicación están disponibles en JSONStore, incluso cuando el dispositivo que ejecuta la aplicación no esté en línea. Este persistente almacenamiento, siempre está disponible y puede ser útil para los clientes, empleados o socios, para darles acceso a los documentos cuando, por ejemplo, no hay ninguna conexión de red al dispositivo” (IBM, 2013)

Esta API esta solo disponible para el desarrollo de aplicaciones con sistema operativo Android, iOS, Windows 8 y Windows 8 Phone.

- **Funciones**

- Es una API que se muestra amigable con el desarrollador proporciona a los desarrolladores la capacidad de poblar el almacén de documentos local, y permite actualizar, eliminar y buscar a través de los documentos.
- La persistencia del almacenamiento, basado en archivos coincide con el alcance de la aplicación.
- Los datos almacenados se cifran con el cifrado AES 256, lo que proporciona seguridad y confidencialidad. Se da un segmento de para el usuario ya que posee seguridad por contraseña, lo que permite que se pueda acceder más de un usuario en el dispositivo.
- Capacidad para realizar un seguimiento de cambios locales.



Ilustración 31 Representación gráfica básica de JSONStore

Fuente: IBM Worklight Foundation

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

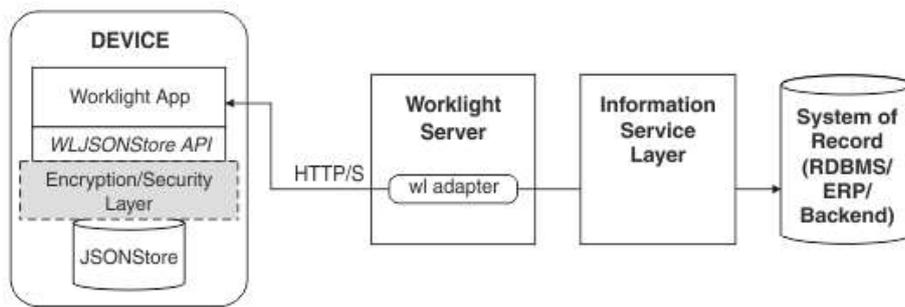


Ilustración 32 Componentes en interacción de JSONStore

Fuente: IBM Worklight Foundation

• **Comparativa**

JSONStore es una API de JavaScript para el almacenamiento de datos dentro de las aplicaciones híbridas, una API de Objective-C para aplicaciones iOS nativas, y una API de Java para aplicaciones nativas de Android. Como referencia, aquí hay una comparación de las diferentes tecnologías de almacenamiento de JavaScript para ver cómo JSONStore compara con ellos.

JSONStore es similar a tecnologías como LocalStorage, indexado DB, Córdoba API de almacenamiento, Córdoba API Archivo y WorklightEncrypted caché. La tabla muestra cómo algunas de las características que son proporcionados por JSONStore compararán con otras tecnologías. La característica JSONStore sólo está disponible en iOS y dispositivos Android y simuladores.

	JSONStore	Encrypted Cache	LocalStorage	IndexedDB	Cordova Storage	Cordova File
Android Support (Hybrid & Native Applications)	✓	✓	✓	✓	✓	✓
iOS Support (Hybrid & Native Applications)	✓	✓	✓	✓	✓	✓
Windows 8 and Windows 8 Phone Support	✓	✓	✓	✓	-	✓
Web	Dev Only (See Note 1)	✓	✓	✓	-	-
Data encryption	✓	✓	-	-	-	-
Maximum Storage	Available Space	~5 MB	~5 MB	>5 MB	Available Space	Available Space
Reliable Storage (See Note 2)	✓	-	-	-	✓	✓
Keep Track of Local Changes	✓	-	-	-	-	-
Multi-user support	✓	-	-	-	-	-
Indexing	✓	-	-	✓	✓	-
Type of Storage	JSON Documents	Key/Value Pairs	Key/Value Pairs	JSON Documents	Relational (SQL)	Strings

Ilustración 33 Comparación entre tecnologías de almacenamiento de datos

Fuente: IBM Worklight Foundation

3.4. HERRAMIENTAS

IBM Worklight Studio al igual que otras plug-in de desarrollo, posee algunas herramientas propias de la misma, para que puedan ser utilizadas por lo desarrolladores, y de esa manera optimicen el desarrollo de su proyecto.

Las herramientas que posee IBM Worklight Studio ayudan con la implementación del código en tiempo real.

3.4.1. Rich Page Editor

El Rich Page editor es un editor para páginas HTML, además mediante él se puede añadir widgets de Dojo, como crear y editar páginas para dispositivos móviles.

Es un editor que posee dentro de ellas pestañas para ver la codificación del proyecto móvil, en tres tipos de vistas o representaciones para la página.

- **Vistas**

Como se había dicho antes Rich Page Editor permite visualizar el código tal y como, como también la vista del diseño que se muestra tal y como saldrá en el browser y también la división de las 2.

Cada vista posee las siguientes funciones adicionales que permite que esta herramienta haga más rápido el diseño de la aplicación.

- Navegación Móvil
- Outline⁴⁴
- Vista de Propiedades
- Barra de herramientas
- Barra de Menú
- Menú Pop-up⁴⁵
- Paleta de Componentes

⁴⁴ **Outline:** es un esquema que se usa para tomar notas.

⁴⁵ **Pop-up:** El término denomina a las ventanas que emergen automáticamente (generalmente sin que el usuario lo solicite).

VISTAS DE EDICIÓN	DESCRIPCIÓN
Código	Permite ver el trabajo directamente del código fuente que se está creando tanto el código que el desarrollador genera manualmente como de las paletas y demas.
Compartida	La vista compartida muestra el código y el diseño. Se puede visualizar los cambios en el diseño si el código se cambia. Mediante la paleta se puede controlar otros mecanismos, como si la vista que se desee sea horizontal o vertical. 
Diseño	El diseño de la vista nuestro por completo el diseño de la aplicación y se pueden mover los elementos que se deseen, como también mediante el menú de herramientas se puede seguir anclando más elementos. Al realizar o añadir algún cambio en el diseño directamente el código se va alterando en los cambios que se produjeron. Además mediante la paleta se puede simular el diseño en diferentes dispositivos.

Tabla 1 Vistas del Rich Page Editor

Fuente: IBM Worklight Foundation

3.4.2. Mobile Browser Simulator

Es una aplicación web que permite simular en tiempo real la aplicación que se está desarrollando en IBM Worklight Studio, la ventaja de esta es que es propia de IBM Worklight y no necesita instalar que instalar ningún SDK, como lo hacen otros software de desarrollo móvil.

El simulador también permite cargar skins, además en caso que se edite alguna parte del proyecto, al guardarlo los cambios ya fueron guardados en el simulador, por lo que solo se necesita refrescar la pantalla del simulador.

Para poder ver el simulador es necesario que se tenga instalado Browser como Chrome, Firefox y Safari, preferiblemente últimas versiones.

“El simulador contiene un marco que emula un dispositivo final. Le muestra lo que su página se ve, como en el interior del navegador del dispositivo móvil. Puede cambiar el marco de emular diferentes resoluciones de pantalla y formar factores, incluyendo BlackBerry 6 y 7, BlackBerry 10, Android, iPad, iPhone, teléfono y 8 dispositivos móviles de Windows. También puede girar el marco para el cambio de orientación (vertical u horizontal). Puede agregar varios dispositivos en el frame para ver las

diferentes pantallas simultáneamente. Posee un servlet de detección de dispositivo que está configurado para su proyecto web, el simulador emula las peticiones de los diferentes agentes específicos del dispositivo.” (IBM, 2013)

3.5. DESARROLLO DE APLICACIONES

IBM Worklight Studio es una herramienta que permite el desarrollo de aplicaciones móviles, para tablets, celulares, etc. La ventaja de esta herramienta es que puede realizar en desarrollo multiplataforma.

Aunque IBM Worklight Studio pueda desarrollar aplicaciones, la misma posee algunas diferencias en su desarrollo, por lo que es necesario analizar cada una de ellas.

En esta sección de este capítulo pretende dar a conocer en general el desarrollo de aplicaciones que se realizan dentro de IBM Worklight Studio, el objetivo de esta sección es proporcionar un contenido del desarrollo de aplicaciones muy comprensibles en terminología y en funciones que realiza internamente esta herramienta.

Es claro recalcar que dentro de esta sección no se realizará un desarrollo de un proyecto completo solo se analizará las partes del mismo.



Ilustración 34 Simulador Móvil de IBM Worklight

Fuente: IBM Worklight Foundation

3.5.1. Proceso de desarrollo de una aplicación móvil

IBM Worklight posee un proceso con el que se crea y desarrolla cualquier tipo de aplicación móvil dentro de su plataforma.

El proceso es el siguiente:

“En un nivel superior, las tareas de flujo de trabajo que sigue cuando desarrolla una aplicación móvil son similares a las del desarrollo de una aplicación tradicional. Una vista simplificada de los pasos, podría ser así:

1. Configuración de Worklight y comenzar a desarrollar la aplicación.
2. Proceda con el desarrollo usual y el proceso de prueba para verificar la funcionalidad de la aplicación al ejecutar o simular la aplicación.
3. Cuando piense que está listo, publique la aplicación de prueba para obtener comentarios de los usuarios clave.
4. Según los comentarios, continúe trabajando en las mejoras de la aplicación o láncela para una audiencia de prueba más amplia.
5. Si decide expandir el uso de la aplicación de prueba, puede publicar la aplicación para un grupo de usuarios restringido en un repositorio local o privado de la aplicación, o para un repositorio público o empresarial de la aplicación (como Apple App Store, Google Play y demás).” (IBM, 2013)

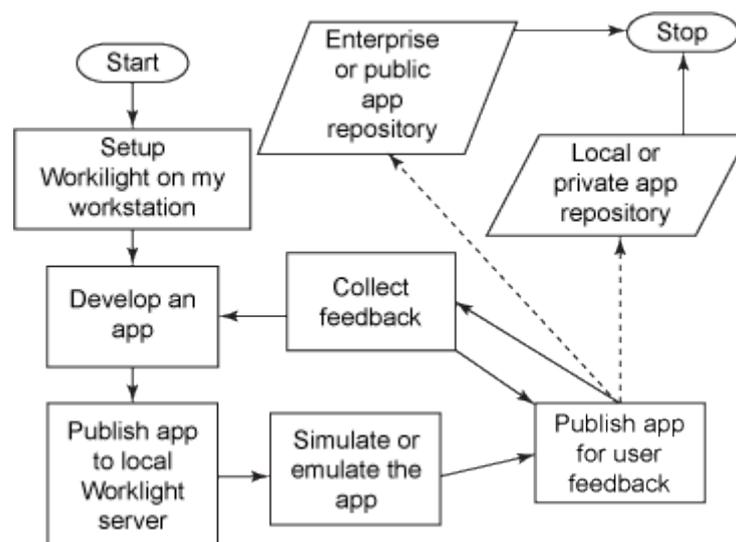


Ilustración 35 Desarrollo de una aplicación móvil

Fuente: IBM Worklight Foundation

3.5.2. Desarrollo de aplicaciones web e híbridas

Para este tipo de aplicaciones el desarrollo es el similar, ya que no se requieren de funciones nativas para su creación.

- **Anatomía de un Proyecto Worklight**

Al crear un proyecto como ya se lo había indicado antes, se crean una serie de carpetas que ayudan con la organización adecuada del proyecto, en algunas de ellas se puede ver que contienen archivos de configuración y creación por defecto.

<project-name>	Root project folder
adapters	Source code for all adapters belonging to the project
apps	Source code for all applications belonging to the project
bin	Artifacts resulting from building adapters, apps, and server-side configuration and libraries
components	Source code for all shell components belonging to the project
www	Source code of the Dojo JavaScript framework, if installed as part of Worklight Studio
server	
conf	Worklight Server configuration files, such as worklight.properties and authenticationConfig.xml
java	Java™ code that must be compiled and packaged into jar files deployable to the Worklight Server
lib	Pre-compiled libraries that must be deployed to the Worklight Server
services	Description, at development stage, of back-end services discovered for consumption by the applications in the project

Ilustración 36 Anatomía de un Proyecto Worklight

Fuente: IBM Worklight Foundation

- **Anatomía de una Aplicación Worklight**

IBM Worklight Studio como se había explicado antes dentro de un proyecto puede contener muchas aplicaciones, y estas a su vez pueden ser de diferentes ambientes.

Al crear una aplicación web se crean dentro del proyecto carpetas, y esto mismo se realiza por cada aplicación creada.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

<app-name>	Main application folder
common	Application resources common to all environments
css	Style sheets to define the application view
images	Thumbnail image and default icon
js	JavaScript files
index.html	An HTML5 file that contains the application skeleton
android	Web and native resources specific to Android
blackberry10	Web and native resources specific to BlackBerry 10
blackberry	Web and native resources specific to BlackBerry 6 and 7
ipad	Web and native resources specific to iPad
iphone	Web and native resources specific to iPhone
windowsphone8	Web and native resources specific to Windows Phone 8
air	Resources specific to Air
desktopbrowser	Resources specific to desktop browsers
mobilewebapp	Web resources specific to mobile web applications
windows8	Resources specific to Windows 8
legal	License documents for the application or third-party software used in the application
application-descriptor.xml	
build-settings.xml	

Ilustración 37 Anatomía de una Aplicación Worklight

Fuente: IBM Worklight Foundation

- **Desarrollo**

El desarrollo de aplicaciones se puede dar de acuerdo a los sistemas operativos a los que se desea desarrollar, IBM Worklight Studio posee ambientes para iOS, Android, BlackBerry y Windows Phone.

Al momento de cargar cualquiera de estas se cargan ambientes para cada una de ellas, al momento de crear la aplicación con cualquier ambiente se crean carpetas por defecto, y a la vez función que cada uno de ellas posee, en algunos ambientes se añaden otras más como recursos multimedia, y configuraciones especiales de cada una.

En esta parte se definirá los conceptos generales del desarrollo de todas las aplicaciones en general.

- **Proceso Inicial por defecto**

Es un proceso es el que se llama al inicio y se crea al momento de creación del proyecto. Lo que se realiza es lo siguiente:

“se llama al initializeWebFrameworkWithDelegate para iniciar el proceso de inicialización. Tras el proceso de inicialización, se llama al

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

wlInitWebFrameworkDidCompleteWithResult. En caso de éxito, el wlInitWebFrameworkDidCompleteWithResult crea el ViewController de Córdoba y carga el archivo HTML principal (a través mainHtmlFilePath) en el WebView de Córdoba. En caso de fallo, se muestra un diálogo de error.”
(IBM, 2013)

The appName.m file

```
@interface Compatibility54ViewController : UIViewController
@end

@implementation Compatibility54ViewController
/**
 In iOS 5 and earlier, the UIViewController class displays views in portrait mode only. To support additional orientations, you must override the
 shouldAutorotateInterfaceOrientation: method and return YES for any orientations your subclass supports.
 */
- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
    return YES;
}
@end

@implementation AppDelegate
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    BOOL result = [super application:application didFinishLaunchingWithOptions:launchOptions];

    // A root view controller must be created in application:didFinishLaunchingWithOptions:
    self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen] bounds];
    UIViewController* rootViewController = [[Compatibility54ViewController alloc] init];

    [self.window setRootViewController:rootViewController];
    [self.window makeKeyAndVisible];

    [[WL sharedInstance] showSplashScreen];

    [[WL sharedInstance] initializeWebFrameworkWithDelegate:self];

    return result;
}

// This method is called after the WL web framework initialization is complete and web resources are ready to be used.
- (void)wlInitWebFrameworkDidCompleteWithResult:(WLWebFrameworkInitResult *)result
{
    if ([result statusCode] == WLWebFrameworkInitResultSuccess) {
        [self wlInitDidCompleteSuccessfully];
    } else {
        [self wlInitDidFailWithResult:result];
    }
}

- (void)wlInitDidCompleteSuccessfully
{
    UIViewController* rootViewController = self.window.rootViewController;

    // Create a Cordova View Controller
    CDVViewController* cordovaViewController = [[CDVViewController alloc] init];

    cordovaViewController.startPage = [[WL sharedInstance] mainHtmlFilePath];

    // Adjust the Cordova view controller view frame to match its parent view bounds
    cordovaViewController.view.frame = rootViewController.view.bounds;

    // Display the Cordova view
    [rootViewController addChildViewController:cordovaViewController];
    [rootViewController.view addSubview:cordovaViewController.view];
}

- (void)wlInitDidFailWithResult:(WLWebFrameworkInitResult *)result
{
    UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"ERROR"
        message:[result message]
        delegate:self
        cancelButtonTitle:@"OK"
        otherButtonTitles:nil];

    [alertView show];
}

```

Ilustración 38 Controlador de eventos en Worklight

Fuente: IBM Worklight Foundation

- **Implementación de un proceso Inicial**

Con esta implementación se demuestra la flexibilidad de la arquitectura Worklight.

La aplicación se inicia desde una página nativa que ejecuta el código personalizado.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

- El usuario hace clic en un botón para inicializar el framework de IBM Worklight Studio.
- Un segundo botón se activa sólo cuando la inicialización se realiza correctamente, y carga la página web de Córdoba con el archivo HTML principal de la aplicación.

Procedimiento

El procedimiento que sigue la implementación de la aplicación es la siguiente.

1. Crea la aplicación híbrida, también se crea automáticamente el archivo .h que extiende de WAppDelegate.

```
@interface MyAppDelegate : WAppDelegate
{
}
@end
```

2. Crea un vista nativa

```
@implementation MyAppDelegate
// Create a custom view before initializing the Worklight framework
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    BOOL res = [super application:application didFinishLaunchingWithOptions:launchOptions];

    // code for building a window

    // Build custom native UI
    MyCustomViewController* customViewController = [[MyCustomViewController alloc] init];

    // Set root view controller
    self.window.rootViewController = customViewController;

    // Show window
    [self.window makeKeyAndVisible];

    return res;
}
@end
```

3. Crea la vista de controlador, y realiza una lógica empresarial personalizada que contiene dos botones:
 - uno para activar la inicialización de Worklight
 - uno para cargar la Webview de Cordova.

Se usa un delegate⁴⁶ para controlar las acciones de la UI⁴⁷, y responder a la realización de inicialización.

```
@interface MyCustomViewController : UIViewController
{
}
@property (retain, nonatomic) MyWLDelegate* wlDelegate;
@end

@implementation MyCustomViewController
//
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    init wlDelegate
    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    //Run some business logic here
}

//Run some more business logic

// Start Worklight initialization
- (IBAction)onInitWLClicked:(id)sender {
    [[WL sharedInstance] initializeWebFrameworkWithDelegate:self.wlDelegate;]}

// Load the Cordova web view
- (IBAction)onShowWebViewClicked:(id)sender {
    [self.wlDelegate showCordovaWebView];
}
}
@end
```

4. Se crea un delegado por defecto que implemento WLInitWebFrameworkDelegate.

```
@interface MyWLDelegate : NSObject <WLInitWebFrameworkDelegate>
//
@end
```

⁴⁶ **Delegate:** es una técnica en la que un objeto de cara al exterior expresa cierto comportamiento pero en realidad delega la responsabilidad de implementar dicho comportamiento a un objeto asociado en una relación inversa de responsabilidad.

⁴⁷ **UI:** Interfaz de Usuario

```
@implementation MyWLDelegate

// Implement WLInitWebFrameworkDelegate protocol
-(void)wlInitWebFrameworkDidCompleteWithResult:(WLWebFrameworkInitResult *)result
{
    if ([result statusCode] == WLWebFrameworkInitResultSuccess) {
        [self wlInitDidCompleteSuccessfully];
    } else {
        [self wlInitDidFailWithResult:result];
    }
}

-(void)wlInitDidCompleteSuccessfully{
    // Enable the button on the viewController
}

// load and show the Cordova WebView
-(void)showCordovaWebView
{
    // build the cordova view controller
    CDWViewController* cordovaViewController = [[[CDWViewController alloc] init]];

    // Set URL of main HTML file
    cordovaViewController.startPage = [[WL sharedInstance] mainHtmlFilePath];

    // Show the Cordova web view
    UIViewController* rootViewController = self.window.rootViewController;
    [rootViewController addChildViewController:cordovaViewController];
    [rootViewController.view addSubview:cordovaViewController.view];
}
}
```

3.5.3. Desarrollo de aplicaciones nativas

El desarrollo de aplicaciones nativas iniciar normalmente con la creación de un proyecto y la de una aplicación.

IBM Worklight Studio tiene integrado un API Nativo, y lo único que se necesita para poder usarlo es al momento de crear una aplicación cambiar en tipo como Native API.

De acuerdo a cada ambiente en el que se desea desarrollar se escoge la API nativa.

- El API de Objective-C, si la aplicación a desarrollar es para el entorno iOS
- El API de Java, si la aplicación a desarrollar es para Android y Java Micro Edition
- El API de C#, si la aplicación a desarrollar es para Windows Phone 8

Al momento de crear la aplicación nativa, se añaden dos archivos importantes para el desarrollo de la aplicación, y estos son los siguientes.

- El archivo descriptor: es el archivo `application-descriptor.xml` que está en el directorio raíz de la aplicación.
- La biblioteca nativa y el archivo de propiedades de cliente: El nombre y el formato de este contenido dependen del ambiente en que se trabaje.
 - **iOS:**
 - La biblioteca es el `WorklightAPI`.
 - El archivo de propiedades del cliente es el `worklight.plist`

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

- **Android:**
 - La biblioteca es el `worklight-android.jar`
 - El archivo `wlclient.properties` es el archivo de propiedades de cliente.
- **Java ME:**
 - La biblioteca es el `worklight-javame.jar` más el archivo `json4javame.jar` juntas definen la biblioteca nativas.
 - El archivo `wlclient.properties` es el archivo de propiedades de cliente.
- **Windows Phone 8:**
 - La biblioteca es `worklight-windowsphone8.dll`. El archivo `Newtonsoft.Json.WindowsPhone.dll` es una biblioteca que se requiere para el uso de objetos JSON en C #.
 - El archivo de propiedades de cliente es `wlclient.properties`.

A más de poseer las API generadas por IBM Worklight Studio, es necesario descargar otros proyectos propios de cada ambiente a desarrollar.

- Un proyecto en el IDE Xcode, para desarrollar una aplicación nativa con Objective-C para el entorno iOS
- Un proyecto con el IDE Eclipse, para desarrollar una aplicación nativa con Java, para entornos Android o para Java ME
- Un proyecto en Visual Studio Express para Windows Phone o Visual Studio 2012 Professional o superior, para Windows Phone 8.

Una vez ya creada la aplicación nativa se debe definir ciertos parámetros para que en desarrollo de la aplicación no se presenten problemas.

- Definir los diferentes aspectos de la aplicación estableciendo los valores adecuados en el archivo descriptor de la aplicación.
- Actualizar el archivo de propiedades de cliente, según sea necesario.
- Copiar el archivo de propiedades de cliente y la biblioteca nativa en la ubicación adecuada para el proyecto original. También es necesario crear referencias del proyecto de aplicación nativa.

- **Añadir capacidades a una aplicación nativa existente**

Las capacidades que posee IBM Worklight Studio son muchas, pero en particular la que más puede interesar al momento de crear un aplicación nativa, es que también se la puede convertir en híbrida.

El Proceso es el siguiente:

- **Exportar los recursos correspondientes de un proyecto existente.**

Utilizando el plug-in de Eclipse o la línea de comandos, se puede exportar los recursos correspondientes a partir de un proyecto híbrido existente. La primera vez que la adición de la vista web para una aplicación nativa, debe seleccionar la opción de exportar las bibliotecas nativas.

Las Exportaciones posteriores no necesitan incluir estas bibliotecas y resultarán en archivos mucho más pequeños.

- **Integrar los recursos en su aplicación nativa y actualizar su código para mostrar una vista web.**

Integrar los recursos híbridos en la aplicación nativa para mostrar una vista web de Worklight. Este paso es diferente de acuerdo al ambiente como puede ser para Android, iOS y Windows Phone 8.

- **Actualizar el código Nativo**

La principal razón para actualizar una aplicación se debe probablemente a que la aplicación utiliza código nativo y se quiera agregar nuevas características o entregar soluciones que requieren cambios en el código nativo.

No se puede utilizar el mecanismo de actualización directa en cualquiera de estos casos, por lo que se tiene que construir y desplegar una nueva versión de la aplicación; para esto es necesario realizar la actualización mediante el centro de aplicaciones.

Centro de aplicaciones

El centro de aplicaciones se instala al mismo tiempo que IBM Worklight Studio, dentro de este centro se guardan actualizaciones de todo componente, y también se encarga de dar notificaciones sobre actualizaciones.

Cada actualización para cada componente se hace de manera diferente, por lo que es necesario leer especificaciones de la misma en caso que se requiera realizar una actualización.

3.6. OPTIMIZACIÓN Y ACELERACIÓN DE COMPONENTES

En la programación la reutilización de código es indispensable para cualquier proyecto, porque hace que la ejecución del proyecto sea más rápido, y evita que se haga código repetible.

IBM Worklight Studio también posee la misma lógica es por eso que hay optimización y aceleración de componentes para que las aplicaciones que se desarrollen más rápido.

3.6.1. Aceleración de componentes

Los componentes de aplicación son las bibliotecas reutilizables que se pueden agregar a las aplicaciones que se desarrollan. Un componente de aplicación puede ser una biblioteca del lado del cliente o un bloque de tiempo de ejecución del servidor. Bibliotecas típicas podrían manejar las funciones básicas como login o pagos. También pueden contener diversos elementos, tales como objetos no visuales de tiempo de ejecución, los componentes visuales, adaptadores de integración, y paquetes de pantalla de la interfaz de usuario.

Una aplicación de banca puede requerir una biblioteca de procesamiento de imagen para el procesamiento de cheques, un objeto de tiempo de ejecución no visual, y un adaptador de integración para conectar con el sistema bancario para la verificación. Un desarrollador podría considerar el montaje de estos bloques de construcción reutilizables en componentes de la aplicación y, a continuación, agregarlos a múltiples proyectos Worklight® para acelerar el desarrollo de aplicaciones para una gama de diferentes dispositivos.

“Un componente de aplicación puede ayudar a simplificar y acelerar la entrega de aplicaciones móviles de alta calidad a través de múltiples dispositivos. Un componente de la aplicación también puede ayudar a los desarrolladores en sus interacciones con los clientes, puede proporcionar servicios de valor añadido, y puede ayudar a los desarrolladores a entender cómo los consumidores utilizan sus aplicaciones móviles.”
(IBM, 2013)

- **Creación de componentes**

Si se desea crear un componente, lo único que se necesita hacer es crear el proyecto y luego crear un componente de aplicación.

Luego se procede a llenar la metadata de la aplicación de componente y se define el tipo de componente de aplicación que se desea crear.

Field	Description
Name	Name of the application component. Spaces are allowed in this field.
ID	Unique identifier for the application component. This is a read-only field. The identifier is the combination of information specified in the Name field (using upper case characters and without spaces) together with unique identifiers.
Author	Author or provider of the application component.
Version	Version of the application component expressed in VRM (version, release, modification) format; for example, 1.0.1.
Description	Short description of the application component.
Image	Thumbnail image that represents the application component. Supported resolution: 15x12 pixels.

Ilustración 39 Componentes Metadata de la Aplicación

Fuente: IBM Worklight Foundation

- **Visualización de componentes**

Después de crear un componente se puede ver su estructura mediante la herramienta compresión de archivos.

El componente de aplicación contiene carpetas en función de los recursos del proyecto Worklight® que fueron seleccionadas cuando se creó el componente de aplicación, así como una carpeta COMPONENT-DATA obligatorio.

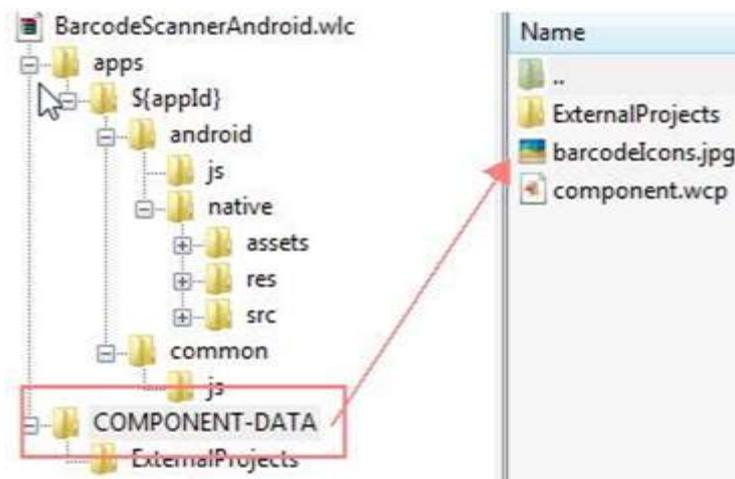


Ilustración 40 Estructura típica de una carpeta de un componente de aplicación

Fuente: IBM Worklight Foundation

La carpeta COMPONENT-DATA contiene los siguientes archivos:

- El thumbnail⁴⁸ de imagen que se seleccionó al crear el componente de aplicación.
- Un archivo de componente llamado component.wcp, que contiene la información de metadatos que se especificó cuando se creó el componente de aplicación.

```
<ComponentData>
  <ID>BarCodeScannerUniqueID</ID>
  <Name>Barcode Scanner Android</Name>
  <Author>IBM</Author>
  <Description>Barcode Scanner for Android by IBM</Description>
  <Version>1.0.0</Version>
  <Image>barcodeIcons.jpg</Image>
  <WLVersion>6.1.0</WLVersion>
</ComponentData>
```

Ilustración 41 Estructura del archivo component.wcp

Fuente: IBM Worklight Foundation

3.6.2. Optimización de componentes

Worklight Studio tiene varias características que se pueden utilizar para reducir el tamaño de una aplicación o de otra manera mejorar su rendimiento o reducir su tiempo de carga.

Durante el desarrollo, las aplicaciones pueden funcionar bien, pero cuando estas aplicaciones son utilizadas por los dispositivos móviles, el rendimiento puede verse afectado por una serie de factores.

El gran tamaño de las aplicaciones puede hacer que el tiempo de descarga desde el Centro de Aplicaciones sea demasiado para los usuarios. La inclusión de múltiples archivos JavaScript en el navegador y aplicaciones web móviles pueden requerir múltiples peticiones para recuperar datos, esto hace que se alargue el tiempo de respuesta. Los recursos no utilizados como imágenes de gran tamaño o archivos innecesarios incluidos en el archivo de caché generado puede hacer que el tiempo de inicialización de la aplicación sea más lento.

⁴⁸ **Thumbnail:** son imágenes en miniatura usadas para ayudar a su organización y reconocimiento. En la era de las imágenes digitales, los motores de búsqueda visuales y los programas para organizar imágenes.

Worklight Studio posee una serie de características que pueden reducir el tamaño de trabajo de sus aplicaciones, como la minimización o eliminación de componentes no utilizados como JSONStore. También incluye características que pueden mejorar el rendimiento y la satisfacción de los usuarios por lo que les permite iniciar más rápido, como la concatenación y la minificación de la caché Manifiesto.

- **Inclusión y exclusión de funciones de la aplicación**

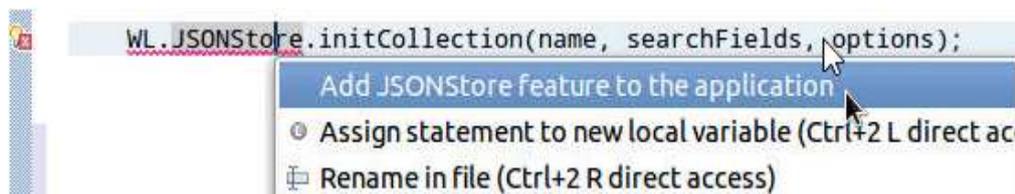
Si no se utilizan algunas características de JSONStore en una aplicación o en ciertos ambientes, se puede reducir el tamaño de la aplicación al excluirlas.

IBM Worklight Studio, puede incluir o excluir las características de construcción de la aplicación, si estas no son necesarias. Por ejemplo, JSONStore ofrece muchos beneficios, si el código que hace referencia se utiliza realmente en la aplicación. Si no se utiliza, los recursos JSONStore aumentan considerablemente como el tamaño de la aplicación, por lo tanto lento tanto la descarga de la aplicación como su inicialización.

Hay un nuevo <feature> en el descriptor de la aplicación que controla la inclusión o exclusión de los recursos. Este se encuentra dentro del application-descriptor.xml como lo siguiente:

```
<application xmlns="http://www.worklight.com/application-descriptor" id="myApp" platformVersion="6.0.0">
...
  <features>
    <JSONStore/>
  </features>
...
</application>
```

Hay crear el <feature>, se pueden tener errores al ejecutar la aplicación en donde se llama a JSONStore, pero esta se puede solucionar de la siguiente manera.



- **Gestión de la caché de aplicaciones web y móviles web**

Worklight® Studio proporciona mecanismos por los cuales usted puede controlar el contenido de la caché.

El caché de la aplicación

Lo ideal es que las aplicaciones móviles y web aun así se pierdan la conexión de la red. Los navegadores más antiguos tenían sus propios mecanismos de caché, pero no siempre eran fiables. El lanzamiento de HTML5 abordó esta necesidad con la introducción de la caché de la aplicación, que ofrece a los usuarios tres ventajas:

- **Navegación fuera de línea** - los usuarios pueden trabajar con la aplicación cuando están fuera de línea.
- **Velocidad** - recursos se encuentran en el caché local, por lo tanto se cargan más rápido.
- Reducción de la carga del servidor - el navegador sólo se descargan los recursos que se actualizan o cambiar desde el servidor.

El manifiesto del caché

El manifiesto de la caché es un archivo de texto que enumera los recursos que el navegador puede almacenar en el caché al trabajar sin conexión. Contiene una lista de los recursos que se almacenan de forma explícita después de la primera vez que se descargan. El manifiesto de caché puede contener tres secciones:

- **CACHE:** archivos y recursos enumerados inmediatamente después de la creación de **CACHE MANIFEST** en él se almacenan todos los archivos que son importantes explícitamente después de la primera vez que se descargan.
- **RED:** Los archivos listados en esta sección son los recursos de la lista blanca que requieren una conexión con el servidor. Todas las solicitudes a estos recursos se guardan en la caché, incluso si el usuario está desconectado.
- **RÉPLICA:** es una sección opcional que especifica las páginas de retorno si un recurso es inaccesible. La primera URI⁴⁹ de la lista es el recurso

⁴⁹ **URI:** Uniform Resource Identifier, es una cadena de caracteres que identifica los recursos de una red de forma unívoca.

principal, y el segundo es el URI de reserva. Ambos URI deben ser relativas y desde el mismo origen que el archivo de manifiesto.

Cuando el navegador abre un documento que incluye el atributo manifest, el navegador carga el documento y luego recupera todas las entradas que aparecen en el archivo de manifiesto de la caché. Si no existe una caché de la aplicación, el navegador crea la primera versión de la caché de la aplicación.

- **Minificación de JS y CSS**

Worklight® Studio permite minimizar el tamaño de los archivos JavaScript y CSS que ya han sido desplegados con el navegador de escritorio y aplicaciones web móviles.

Minificación es el proceso que resta recursos de Internet para hacerlos más pequeños. El menor tamaño de los recursos significa menos tráfico entre la aplicación y del servidor. Se produce un tráfico gigante cuando la aplicación está siendo descargada inicialmente por los usuarios, y al tiempo de inicialización de la aplicación. El uso de estas características puede que mejore el tiempo de inicio de las aplicaciones (concatenación), o reducir el tamaño de la aplicación (minificación).

Minificación se realiza en tiempo de compilación por el Google Closure Compiler⁵⁰. Hay tres niveles de minimización que se pueden utilizar en una aplicación de IBM Worklight Studio, que se enumeran en la siguiente tabla:

Valor	Descripción
None	No se realiza minificación en el código por parte del constructor de Worklight Studio.
whitespaces	Elimina los comentarios del código y también elimina los saltos de línea, espacios innecesarios, y otros espacios en blanco. La salida JavaScript es funcionalmente idéntica a la fuente de JavaScript. (En IBM Worklight Studio para configurar un de estos se va a Build Center Editor -> se busca por Remove whitespaces and comments)
simple	Elimina los espacios en blanco y los comentarios, sino que también optimiza expresiones y funciones, incluyendo el cambio de nombre

⁵⁰ **Google Closure Compiler:** es un conjunto de herramientas para ayudar a los desarrolladores a crear aplicaciones web enriquecidas con JavaScript. Fue desarrollado por Google para el uso en sus aplicaciones web como Gmail, Google Docs y Google Maps.

	<p>de variables locales y parámetros de la función de nombres más cortos. Cambiar el nombre de las variables a nombres más cortos hace el código más pequeño. Debido a la configuración sencilla renombra sólo símbolos que son locales en las funciones, y que no interfiera con la interacción entre el JavaScript compilado y otra JavaScript. Con este ajuste siempre se conserva la funcionalidad de sintácticamente válida JavaScript, si el código no tiene acceso a las variables locales con nombres de cadena, por ejemplo, mediante el uso de eval () declaraciones. (En IBM Worklight Studio para configurar un de estos se va a Build Center Editor -> se busca por Google Closure Compiler Optimización simple.)</p>
--	---

Tabla 2 Opciones para la minificación a nivel de atributo

Fuente: IBM Knowledge Center

Para crear una configuración minificación para aplicación web o móvil. Primero se debe editar las configuraciones generales para la aplicación, y luego activar la minificación para los entornos individuales.

Para configurar minificación, en IBM Worklight Studio, se hace doble clic en el elemento settings.xml de la aplicación luego dentro de la misma se hace clic en Build Settings Editor y se agrega el ambiente y también se escoge el nivel de minificación.

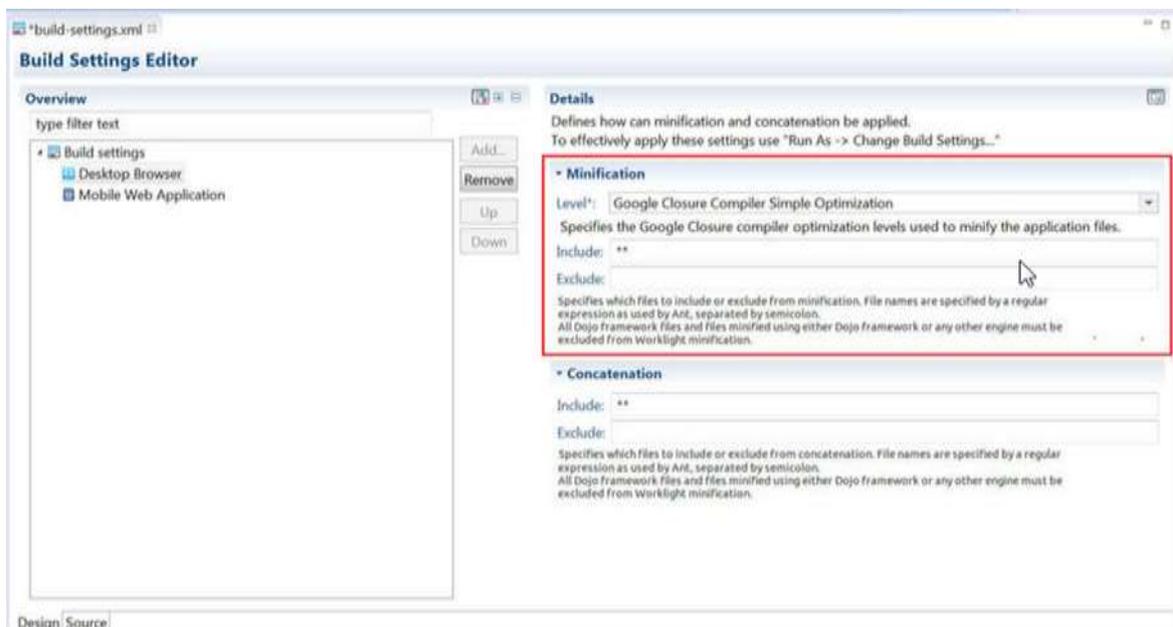


Ilustración 42 Propiedades de Minificación en Worklight

Fuente: IBM Knowledge Center

- **Concatenación de JS y CSS**

Worklight® Studio permite la concatenación de múltiples archivos JavaScript y CSS que se implementan con el navegador y aplicaciones web móviles.

“Desde IBM® Worklight V6.0.0, la función de concatenación permite la concatenación de los múltiples recursos de la web que se utilizan para la aplicación (archivos JavaScript y CSS) en un número más reducido de archivos. La reducción del número total de archivos que hace referencia a los resultados HTML de la aplicación en un menor número de peticiones del navegador cuando se inicia la aplicación, lo que permite que la aplicación se inicie más rápidamente.” (IBM, 2013)

Durante la concatenación, varios recursos (por ejemplo, archivos JavaScript y scripts en línea) se copian en un nuevo archivo, que luego se hace referencia en la aplicación HTML. Las referencias a los recursos originales se eliminan del HTML. Esto significa que se requiere menos comunicación entre el servidor web y el dispositivo para recuperar el código de aplicación.

Al momento de la construcción, el algoritmo concatenación determina qué recursos para concatenar en la que los archivos. La concatenación es controlada por un número de diferentes parámetros, tales como la estructura del HTML, del tipo de los recursos que se concatena, y los atributos de estos recursos. El orden de los recursos en el HTML se conserva. Como resultado, el proceso de concatenación no tiene ningún efecto negativo en términos de dependencias de código o funcionalidad.

Para configurar concatenación, en IBM Worklight Studio, se hace doble clic en el elemento settings.xml de la aplicación luego dentro de la misma se hace clic en Build Settings Editor y se agrega el ambiente.

3.7. ADMINISTRACIÓN Y MONITOREO

IBM Worklight Studio a más de poseer herramientas de desarrollo, implementación, simulación, optimación y otras.

Todo software de desarrollo en la actualidad deben de poseer herramientas que ayuden con la administración de una aplicación y con el monitoreo de cada una de los componentes que posee, esto ayuda a que el servicio que de la aplicación no posee problemas a futuro y se puede controlar posibles inconvenientes para los usuarios.

IBM Worklight Studio proporciona varias maneras de administrar el desarrollo de las aplicaciones en producción. La consola es la principal herramienta con la que puedes controlar todas las aplicaciones desplegadas desde una consola centralizada basada en web.

Las principales operaciones que se pueden realizar a través de la consola de IBM Worklight Studio son:

- Implementar aplicaciones móviles y adaptadores del servidor.
- Administrar versiones de la aplicación para desplegar nuevas versiones o remotamente desactivar las versiones antiguas.
- Administrar los dispositivos móviles y los usuarios para gestionar el acceso a un dispositivo o de acceso específico para un usuario específico a una aplicación.
- Mensajes de notificación de visualización en el inicio de la aplicación.
- Supervisar los servicios de notificación push.
- Recopilar registros del lado del cliente para aplicaciones específicas instaladas en un dispositivo específico.

No todo tipo de usuario de administración puede realizar todas las operaciones de la administración. La consola tiene cuatro roles diferentes definidos para la administración de aplicaciones de IBM Worklight. Las siguientes funciones de administración IBM Worklight Studio se definen:

- **Monitor**

En esta función, el usuario puede supervisar desplegado proyectos faro de trabajo y artefactos desplegados. Este papel es de sólo lectura.

- **Operador**

Un operador puede realizar todas las operaciones de gestión de aplicaciones móviles, pero no puede agregar o quitar versiones de las aplicaciones o adaptadores.

- **Deployer**

En este rol, un usuario puede realizar las mismas operaciones que el operador, pero también puede desplegar aplicaciones y adaptadores.

- **Administrador**

En este rol, un usuario puede realizar todas las operaciones de administración de la aplicación.

Nota: En IBM Worklight Studio V6.2.0, tiene la siguiente configuración de autenticación:

- Rol "worklightadmin", el usuario "admin", contraseña "admin"
- Rol "worklightdeployer", el usuario "deployer", contraseña: "demo"
- Rol "worklightmonitor", el usuario "monitor", contraseña: "demo"
- Rol "worklightoperator", el usuario "operator", contraseña: "demo"

• **Administración de aplicaciones con la consola de IBM Worklight Studio**

A través de la consola de IBM Worklight Studio se pueden administrar aplicaciones ya que puede administrar la implementación de nuevas versiones de las aplicaciones móviles y de escritorio, el bloqueo de aplicaciones o denegar el acceso y mostrar los mensajes de notificación.

Se puede utilizar la consola para ver todas las aplicaciones que se instalan y todas las plataformas de dispositivos que son compatibles. Puede utilizar la consola para desactivar versiones de las aplicaciones específicas de plataformas específicas y obligar a los usuarios a actualizar la aplicación antes de continuar a usarlos.

4. MARCO DE DESARROLLO

4.1. GENERAL

4.1.1. Introducción

Para el desarrollo del aplicativo del proyecto llamado ESTUDIO DE LA HERRAMIENTA IBM WORKLIGHT STUDIO PARA EL DESARROLLO DE APLICACIONES MÓVILES MULTIPLATAFORMA CON APLICATIVO DE CONSULTA DE NOTAS, LIBROS Y SÍLABOS EN EL INSTITUTO TECNOLÓGICO SUPERIOR “JOSÉ CHIRIBOGA GRIJALVA”; como ya se lo había mencionado antes se aplicará la metodología SCRUM, por una metodología ágil y su principal función es tener un avance rápido del proyecto como tener complacido al cliente.

4.1.2. Propósito

Obtener una aplicación móvil de consultas de notas, libros y sílabos del Instituto Tecnológico Superior “José Chiriboga Grijalva” con el que se probará la investigación que se realizó anteriormente y esta sea desarrollada rápidamente para su pronta ejecución.

4.1.3. Alcance

Este proyecto se delimitara a la aplicación de la metodología SCRUM utilizando como herramienta de desarrollo IBM Worklight Studio y la aplicación permitirá realizar las siguientes consultas:

- Consulta de Notas
- Consulta de Bibliografía de Libros o Documentos
- Consulta de Sílabos

4.1.4. Generalidades

La metodología SCRUM reúne ciertos aspectos y características que para el desarrollo de este proyecto se utilizó y cumplió con casi todas de ellas, por lo que es necesario acotar que se omitieron ciertos aspectos que no se consideraron importantes para el

proyecto; pero en si se plasmó la estructura metodológica que debe poseer una proyecto desarrollado por esta metodología.

4.2. COMPONENTES

4.2.1. Roles y Responsabilidades

ROL	RESPONSABILIDAD	RESPONSABLE
SCRUM Master	Verificar la aplicación de la metodología SCRUM en cada paso del proyecto.	Sr. Vadin Villota TESISTA
Product Owner	Verificar que la realización del proyecto sea de agrado para la empresa y cumpla con las expectativas deseadas que el producto debe cumplir.	Ing. Jorge Acosta JEFE DE SISTEMAS DEL INSTITUTO TECNOLÓGICO SUPERIOR “JOSÉ CHIRIBOGA GRIJALVA”
SCRUM TEAM	Desarrollar por completo el producto tomando en cuenta los requisitos del Product Owner y del SCRUM Master	Sr. Vadin Villota TESISTA

Tabla 3 Tabla de Actores por Roles y Responsabilidades

Fuente: Propia

Como el presente proyecto es un trabajo de grado, dos roles importantes para el desarrollo del proyecto son parte del Sr. Vadin Villota quien desarrolla la tesis.

4.2.2. Elementos

Cada Sprint que se estableció para desarrollar este proyecto tiene una duración de 15 días, que en su total tuvo una duración de 3 meses para el desarrollo de todo el proyecto; cada vez que se realice una reunión se obtendrán los siguientes artefactos.

- **Incremento**

Este artefacto tiene la función de que en él se lleve el registro de actividades realizadas, y podrá ser utilizada por cualquier miembro del equipo.

En este artefacto se determinarán los requisitos de acuerdo a cada Sprint.

- **Sprint Backlog**

Este artefacto tiene la función de registrar de forma detallada todo lo correspondiente al Incremento, y finalizada cada Sprint, debe poseer los siguientes elementos: Descripción de la tarea, Responsable, Estado y Observación.

Dichos elementos debe estar dentro de una tabla que permita la mejor visualización de este artefacto.

- **Product Backlog**

Este artefacto tiene la función de seguir con el seguimiento del producto, debe poseer los siguientes elementos: descripción, Sprint, Responsable y Observación.

Dichos elementos debe estar dentro de una tabla que permita la mejor visualización de este artefacto.

4.2.3. Reuniones y Aplicación en el Proyecto

Para este proyecto se realizaron seis Sprint o iteraciones, en la que se reunieron todos los miembros del proyecto.

1. Primer Sprint

1.1. Incremento

ITEM REALIZADO O ACTIVIDAD	OBSERVACIÓN
Asignación de Roles y Responsabilidades	
Análisis de Requerimientos	
Definir las tablas de base de datos a utilizar	No se debe alterar ninguna tabla de la base de datos ITCA_ERP
Diseño o Bosquejo de Interfaz de usuario	
Definir herramientas a utilizar	

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

Definir las iteraciones en total que se van a realizar en el proyecto	
---	--

Tabla 4 Primer Sprint de Desarrollo

Fuente: Propia

1.2. Sprint Backlog

DESCRIPCIÓN DE LA TAREA	RESPONSABLE	ESTADO	OBSERVACIÓN
Asignación de Roles y Responsabilidades	SCRUM Master y Product Owner	Cumplido	
Análisis de Requerimientos	SCRUM Master y Product Owner	Cumplido	
Definir las tablas de base de datos a utilizar	SCRUM Master, Product Owner y SCRUM Team	Cumplido	
Diseño o Bosquejo de Interfaz de usuario	SCRUM Master, y SCRUM Team	Cumplido	
Definir herramientas a utilizar	SCRUM Master, y SCRUM Team	Cumplido	
Definir las iteraciones en total que se van a realizar en el proyecto	SCRUM Master, Product Owner y SCRUM Team	Cumplido	

Tabla 5 Primer Sprint Backlog de Desarrollo

Fuente: Propia

1.3. Product Backlog

DESCRIPCIÓN	SPRINT	RESPONSABLE	OBSERVACIÓN
Asignación de Roles y Responsabilidades	1er Sprint (1ra Iteración)	SCRUM Master y Product Owner	
Análisis de Requerimientos	1er Sprint (1ra Iteración)	SCRUM Master y Product Owner	

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

Definir las tablas de base de datos a utilizar	1er Sprint (1ra Iteración)	SCRUM Master, Product Owner y SCRUM Team	
Diseño o Bosquejo de Interfaz de usuario	1er Sprint (1ra Iteración)	SCRUM Master, y SCRUM Team	
Definir herramientas a utilizar	1er Sprint (1ra Iteración)	SCRUM Master, y SCRUM Team	
Definir las iteraciones en total que se van a realizar en el proyecto	1er Sprint (1ra Iteración)	SCRUM Master, Product Owner y SCRUM Team	

Tabla 6 Primer Product Backlog de Desarrollo

Fuente: Propia

2. Segundo Sprint

2.1. Incremento

ITEM REALIZADO O ACTIVIDAD	OBSERVACIÓN
Instalación y configuración de herramientas(IBM Worklight Studio)	
Realizar conexión con la base de datos del ITCA_ERP	
Desarrollo de la parte de autenticación del usuario estudiante.	
Configuración del Simulador de la Aplicación	Trabajar con el simulador que posee la Herramienta IBM Worklight Studio
Realizar pruebas de ingreso de usuario estudiante para la autenticación.	

Tabla 7 Segundo Sprint de Desarrollo

Fuente: Propia

2.2. Sprint Backlog

DESCRIPCIÓN DE LA TAREA	RESPONSABLE	ESTADO	OBSERVACIÓN
Instalación y configuración de herramientas(IBM Worklight Studio)	SCRUM Team	Cumplido	
Realizar conexión con la base de datos ITCA_ERP	SCRUM Team	Cumplido	
Desarrollo de la parte de autenticación del usuario estudiante.	SCRUM Team	Pendiente	
Configuración del Simulador de la Aplicación	SCRUM Team	Cumplido	
Realizar pruebas de ingreso de usuario estudiante para la autenticación.	SCRUM Team	Pendiente	

Tabla 8 Segundo Sprint Backlog de Desarrollo

Fuente: Propia

2.3. Product Backlog

DESCRIPCIÓN	SPRINT	RESPONSABLE	OBSERVACIÓN
Instalación y configuración de herramientas(IBM Worklight Studio)	2do Sprint (2da Iteración)	SCRUM Team	
Realizar conexión con la base de datos	2do Sprint (2da Iteración)	SCRUM Team	

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

Desarrollo de la parte de autenticación del usuario estudiante.	2do Sprint (2da Iteración)	SCRUM Team	Pendiente para completar en la tercera reunión
Configuración del Simulador de la Aplicación	2do Sprint (2da Iteración)	SCRUM Team	
Realizar pruebas de ingreso de usuario estudiante para la autenticación.	2do Sprint (2da Iteración)	SCRUM Team	Pendiente para completar en la tercera reunión

Tabla 9 Segundo Product Backlog de Desarrollo

Fuente: Propia

3. Tercer Sprint

3.1. Incremento

ITEM REALIZADO O ACTIVIDAD	OBSERVACIÓN
Desarrollo de las consultas de Notas para el estudiante	
Desarrollo de las consultas de Sílabos para el estudiante	Aún faltan datos de prueba para esta consulta.
Pruebas de las consultas en el Simulador de Windows	
Pruebas de las consultas en el Simulador de iOS	
Pruebas de las consultas en el Simulador de Android	

Tabla 10 Tercer Sprint de Desarrollo

Fuente: Propia

3.2. Sprint Backlog

DESCRIPCIÓN DE LA TAREA	RESPONSABLE	ESTADO	OBSERVACIÓN
Desarrollo de la parte de autenticación del usuario estudiante.	SCRUM Team	Cumplido	
Realizar pruebas de ingreso de usuario estudiante para la autenticación.	SCRUM Team	Cumplido	
Desarrollo de las consultas de Notas para el estudiante	SCRUM Team	Cumplido	
Desarrollo de las consultas de Sílabos para el estudiante	SCRUM Team	Cumplido	
Pruebas de las consultas en el Simulador de Windows	SCRUM Team	Pendiente	Faltan realizar las pruebas de Sílabos
Pruebas de las consultas en el Simulador de iOS	SCRUM Team	Pendiente	Faltan realizar las pruebas de Sílabos
Pruebas de las consultas en el Simulador de Android	SCRUM Team	Pendiente	Faltan realizar las pruebas de Sílabos

Tabla 11 Tercer Sprint Backlog de Desarrollo

Fuente: Propia

3.3. Product Backlog

DESCRIPCIÓN	SPRINT	RESPONSABLE	OBSERVACIÓN
Desarrollo de la parte de autenticación del usuario estudiante.	2do Sprint (2da Iteración)	SCRUM Team	Se terminó de realizar con éxito esta actividad
Realizar pruebas de ingreso de usuario estudiante para la autenticación.	2do Sprint (2da Iteración)	SCRUM Team	Se terminó de realizar con éxito esta actividad
Desarrollo de las consultas de Notas para el estudiante	3er Sprint (3ra Iteración)	SCRUM Team	
Desarrollo de las consultas de Sílabos para el estudiante	3er Sprint (3ra Iteración)	SCRUM Team	
Pruebas de las consultas en el Simulador de Windows	3er Sprint (3ra Iteración)	SCRUM Team	Pendiente para completar para la cuarta reunión
Pruebas de las consultas en el Simulador de iOS	3er Sprint (3ra Iteración)	SCRUM Team	Pendiente para completar para la cuarta reunión
Pruebas de las consultas en el Simulador de Android	3er Sprint (3ra Iteración)	SCRUM Team	Pendiente para completar para la cuarta reunión

Tabla 12 Tercer Product Backlog de Desarrollo

Fuente: Propia

4. Cuarto Sprint

4.1. Incremento

ITEM REALIZADO O ACTIVIDAD	OBSERVACIÓN
Desarrollo de las consultas de Libros o Documentos	
Pruebas de la consulta de Libros y documentos en el Simulador de Windows	
Pruebas de la consulta de Libros y documentos en el Simulador de iOS	
Pruebas de la consulta de Libros y documentos en el Simulador de Android	
Comenzar a poner estilo a la Aplicación Móvil en general	Guiarse con los colores de la institución que es verde y azul.

Tabla 13 Cuarto Sprint de Desarrollo

Fuente: Propia

4.2. Sprint Backlog

DESCRIPCIÓN DE LA TAREA	RESPONSABLE	ESTADO	OBSERVACIÓN
Pruebas de las consultas en el Simulador de Windows	SCRUM Team	Cumplido	Las Pruebas realizadas con sílabos fueron exitosas.
Pruebas de las consultas en el Simulador de iOS	SCRUM Team	Cumplido	Las Pruebas realizadas con sílabos fueron exitosas.
Pruebas de las consultas en el Simulador de Android	SCRUM Team	Cumplido	Las Pruebas realizadas con sílabos fueron exitosas.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

Desarrollo de las consultas de Libros o Documentos	SCRUM Team	Cumplido	
Pruebas de la consulta de Libros y documentos en el Simulador de iOS	SCRUM Team	Cumplido	
Pruebas de la consulta de Libros y documentos en el Simulador de Android	SCRUM Team	Cumplido	
Comenzar a poner estilo a la Aplicación Móvil en general	SCRUM Team	Pendiente	Aún no proporcionan las imágenes que desean agregar a la aplicación.

Tabla 14 Cuarto Sprint Backlog de Desarrollo

Fuente: Propia

4.3. Product Backlog

DESCRIPCIÓN	SPRINT	RESPONSABLE	OBSERVACIÓN
Pruebas de las consultas en el Simulador de Windows	3er Sprint (3ra Iteración)	SCRUM Team	Las pruebas que se realizaron salieron bien.
Pruebas de las consultas en el Simulador de iOS	3er Sprint (3ra Iteración)	SCRUM Team	Las pruebas que se realizaron salieron bien.
Pruebas de las consultas en el Simulador de Android	3er Sprint (3ra Iteración)	SCRUM Team	Las pruebas que se realizaron salieron bien.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

Desarrollo de las consultas de Libros o Documentos	4to Sprint (4ta Iteración)	SCRUM Team	
Pruebas de la consulta de Libros y documentos en el Simulador de Windows	4to Sprint (4ta Iteración)	SCRUM Team	
Pruebas de la consulta de Libros y documentos en el Simulador de iOS	4to Sprint (4ta Iteración)	SCRUM Team	
Pruebas de la consulta de Libros y documentos en el Simulador de Android	4to Sprint (4ta Iteración)	SCRUM Team	
Comenzar a poner estilo a la Aplicación Móvil en general	4to Sprint (4ta Iteración)	SCRUM Team	Pendiente para completar para la quinta reunión

Tabla 15 Cuarto Product Backlog de Desarrollo

Fuente: Propia

5. Quinto Sprint

5.1. Incremento

ITEM REALIZADO O ACTIVIDAD	OBSERVACIÓN
Terminar con la parte del diseño de la aplicación móvil	
Implementar la Aplicación Móvil en el Servidor de la Institución	

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

Realizar pruebas de la Aplicación Móvil en las plataformas de Android, iOS y Windows.	
Realizar los manuales de Usuario y Técnico de la Aplicación Móvil	
Planear cronograma de capacitación de usuarios	

Tabla 16 Quinto Sprint de Desarrollo

Fuente: Propia

5.2. Sprint Backlog

DESCRIPCIÓN DE LA TAREA	RESPONSABLE	ESTADO	OBSERVACIÓN
Comenzar a poner estilo a la Aplicación Móvil en general	SCRUM Team	Cumplido	Se logró insertar las imágenes deseadas en la aplicación.
Terminar con la parte del diseño de la aplicación móvil	SCRUM Team	Cumplido	
Implementar la Aplicación Móvil en el Servidor de la Institución	SCRUM Team	Cumplido	
Realizar pruebas de la Aplicación Móvil en las plataformas de Android, iOS y Windows.	SCRUM Team	Cumplido	
Realizar los manuales de Usuario y Técnico de la Aplicación Móvil	SCRUM Team	Pendiente	No alcanzó el tiempo para realizar los manuales.

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

Planear cronograma de capacitación de usuarios	SCRUM Master, Product Owner, SCRUM Team	Pendiente	Aun no coordinan las fechas para la capacitación a los usuarios de
--	---	-----------	--

Tabla 17 Quinto Sprint Backlog de Desarrollo

Fuente: Propia

5.3. Product Backlog

DESCRIPCIÓN	SPRINT	RESPONSABLE	OBSERVACIÓN
Comenzar a poner estilo a la Aplicación Móvil en general	4to Sprint (4ta Iteración)	SCRUM Team	Se terminó de realizar la actividad.
Terminar con la parte del diseño de la aplicación móvil	5ta Sprint (5ta Iteración)	SCRUM Team	
Implementar la Aplicación Móvil en el Servidor de la Institución	5ta Sprint (5ta Iteración)	SCRUM Team	
Realizar pruebas de la Aplicación Móvil en las plataformas de Android, iOS y Windows.	5ta Sprint (5ta Iteración)	SCRUM Team	
Realizar los manuales de Usuario y Técnico de la Aplicación Móvil	5ta Sprint (5ta Iteración)	SCRUM Team	Pendiente para completar para la sexta reunión
Planear cronograma de capacitación de usuarios	5ta Sprint (5ta Iteración)	SCRUM Master, Product Owner y SCRUM Team	Pendiente para completar para la sexta reunión

Tabla 18 Quinto Product Backlog de Desarrollo

Fuente: Propia

6. Sexto Sprint

6.1. Incremento

ITEM REALIZADO O ACTIVIDAD	OBSERVACIÓN
Capacitar a los usuario de la Aplicación Web	
Entregar la Aplicación Móvil a los encargados	El mantenimiento de la aplicación móvil será realizada en general por el Product Owner
Entregar los manuales de la Aplicación Móvil a los encargados	

Tabla 19 Sexto Sprint de Desarrollo

Fuente: Propia

6.2. Sprint Backlog

DESCRIPCIÓN DE LA TAREA	RESPONSABLE	ESTADO	OBSERVACIÓN
Realizar los manuales de Usuario y Técnico de la Aplicación Móvil	SCRUM Team	Cumplido	Se posee los manuales
Planear cronograma de capacitación de usuarios	SCRUM Master, Product Owner, SCRUM Team	Cumplido	Se planeó rápidamente el cronograma de capacitación
Capacitar a los usuarios de la aplicación	SCRUM Master, Product Owner, SCRUM Team	Cumplido	
Entregar la Aplicación Móvil a los encargados	SCRUM Master, Product Owner y SCRUM Team	Cumplido	
Entregar los manuales de la	SCRUM Master, Product Owner y	Cumplido	

Aplicación Móvil a los encargados	SCRUM Team		
-----------------------------------	------------	--	--

Tabla 20 Sexto Sprint Backlog de Desarrollo

Fuente: Propia

6.3. Product Backlog

DESCRIPCIÓN	SPRINT	RESPONSABLE	OBSERVACIÓN
Realizar los manuales de Usuario y Técnico de la Aplicación Móvil	5ta Sprint (5ta Iteración)	SCRUM Team	La actividad fue realizada a tiempo.
Planear cronograma de capacitación de usuarios	5ta Sprint (5ta Iteración)	SCRUM Master, Product Owner y SCRUM Team	La actividad fue realizada a tiempo.
Capacitar a los usuario de la Aplicación Web	6to Sprint (6ta Iteración)	SCRUM Master y SCRUM Team	
Entregar la Aplicación Móvil a los encargados	6to Sprint (6ta Iteración)	SCRUM Master, Product Owner y SCRUM Team	
Entregar los manuales de la Aplicación Móvil a los encargados	6to Sprint (6ta Iteración)	SCRUM Master, Product Owner y SCRUM Team	

Tabla 21 Sexto Product Backlog de Desarrollo

Fuente: Propia

4.3. DESARROLLO DE LA APLICACIÓN

4.3.1. Arquitectura de la solución

La aplicación móvil se desarrolló mediante la herramienta IBM Worklight Studio, por lo que la arquitectura para toda solución o aplicación web es la misma que la se muestra a continuación.

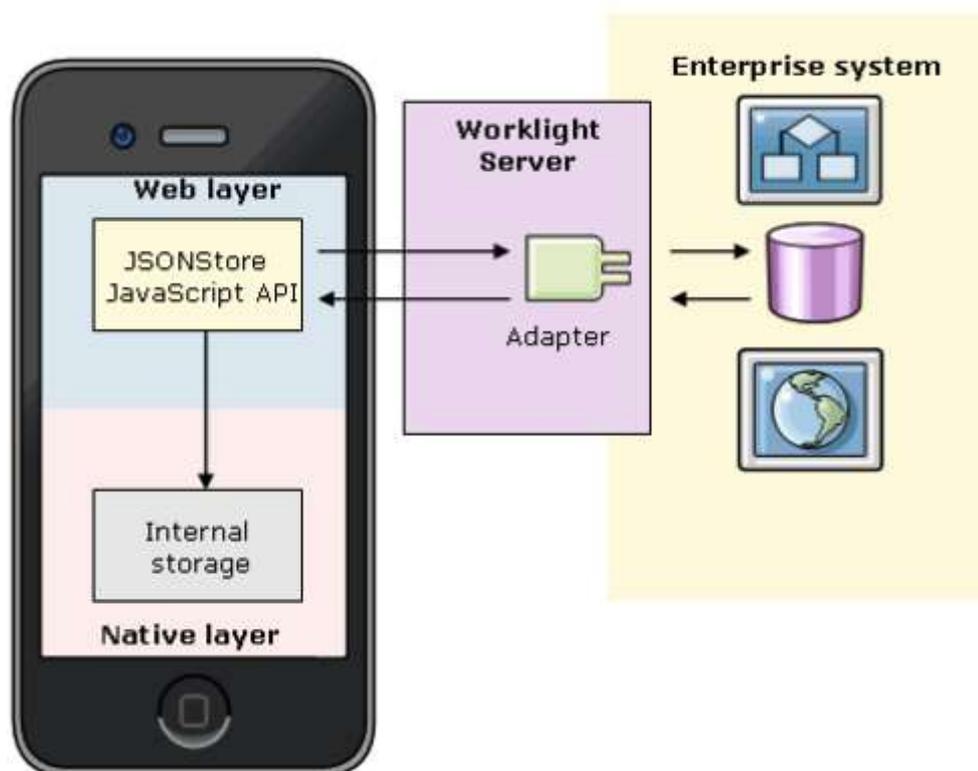


Ilustración 43 Arquitectura de la Aplicación Móvil

Fuente: https://www.ibm.com/developerworks/community/blogs/mobileblog/entry/the_question_of_local_storage7

4.3.2. Base de datos

Aunque en este proyecto no se tuvo necesidad de crear tablas en la base de datos se utilizaron las siguientes tablas para el desarrollo de la aplicación móvil que consistía en consultas.

- **Modelo Físico de las Tablas**

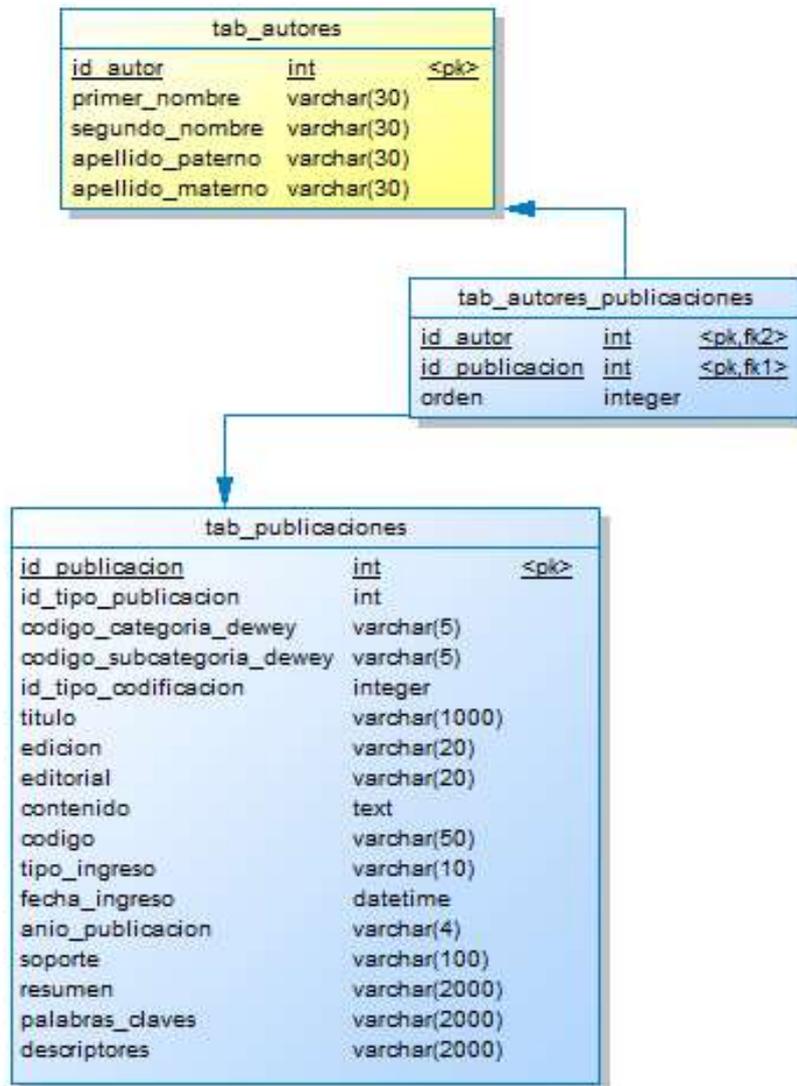


Ilustración 44 Tablas para extraer datos para las búsquedas bibliográficas

Fuente: Propia

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

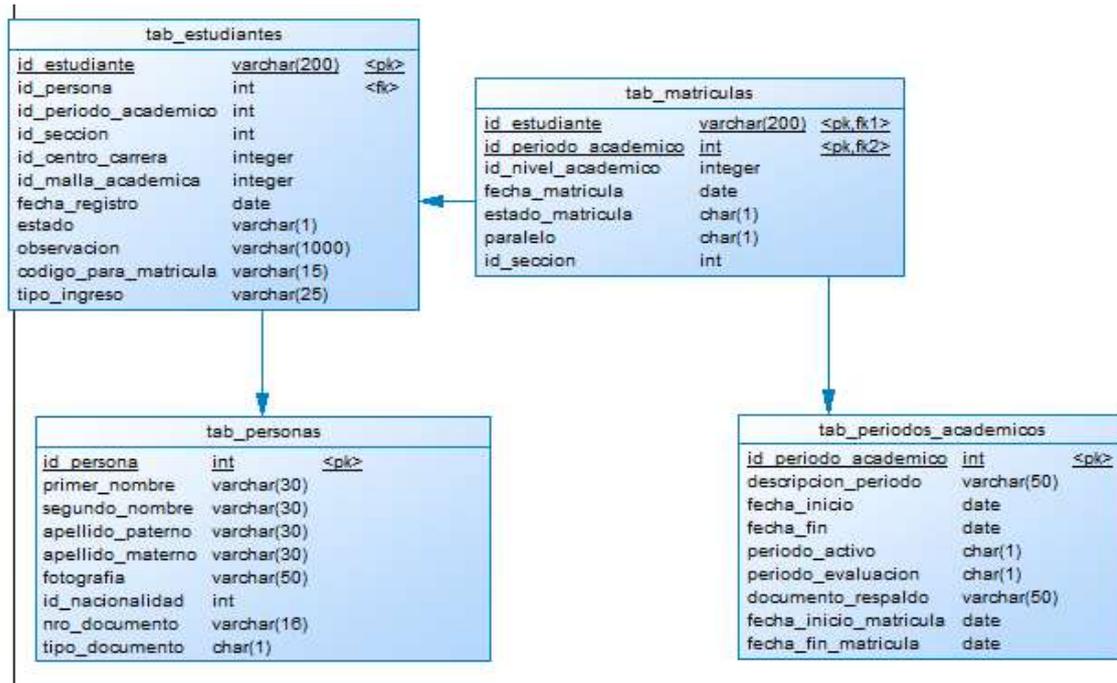


Ilustración 45 Tablas para extraer datos de estudiantes

Fuente: Propia

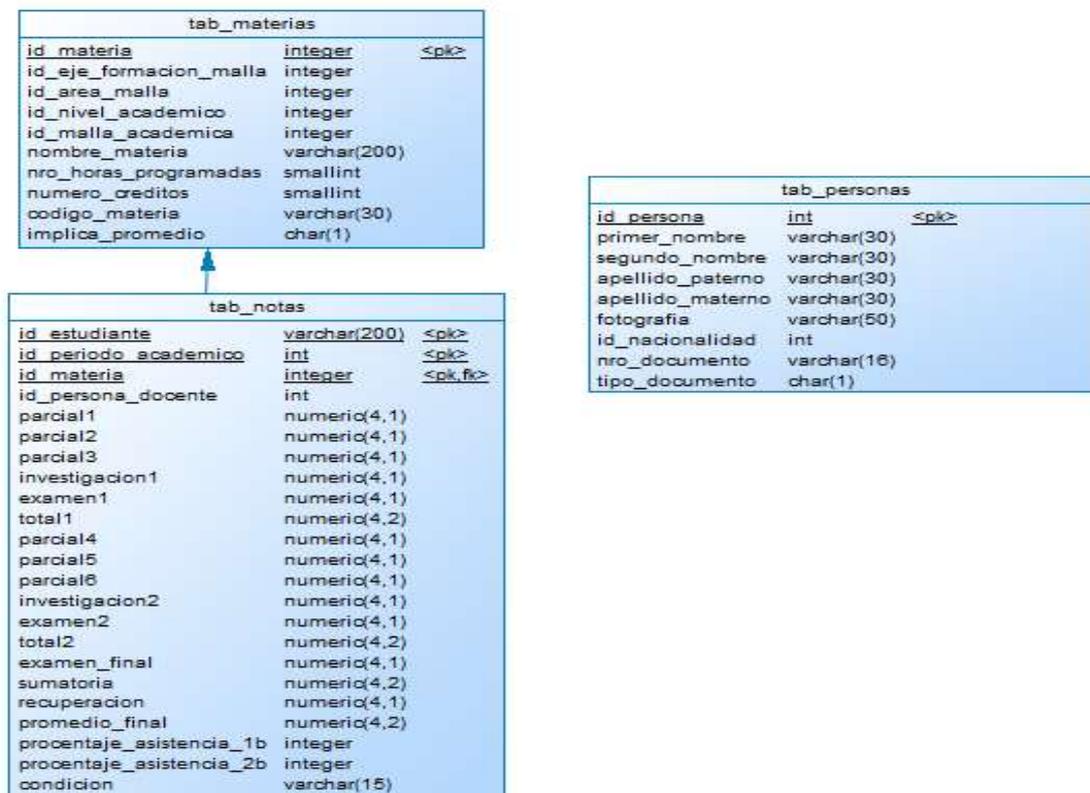


Ilustración 46 Tablas para extraer datos de notas y materias

Fuente: Propia

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

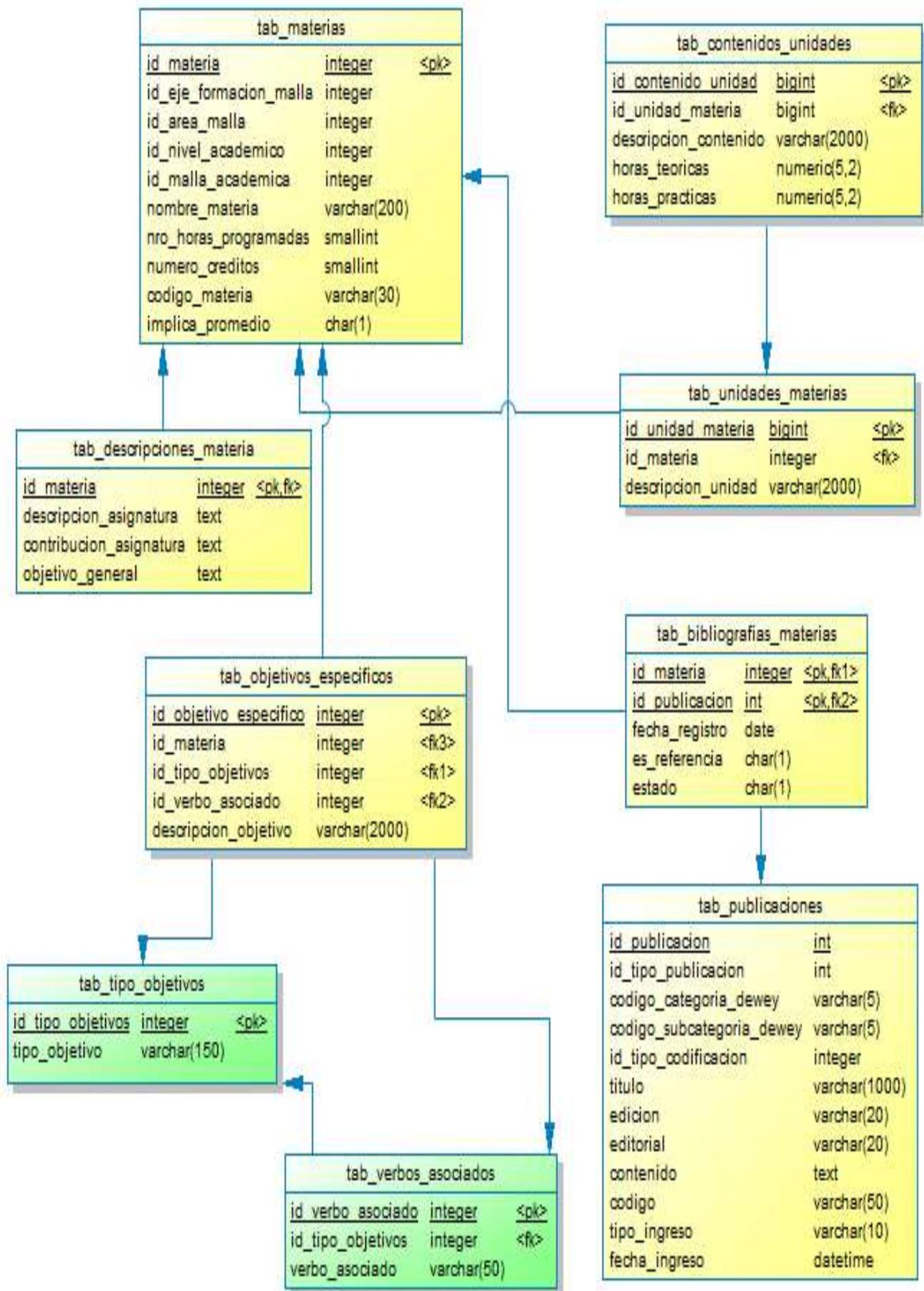


Ilustración 47 Tablas para extraer datos de los sílabos

Fuente: Propia

4.4. INTERFAZ DE USUARIO

El interfaz de usuario se definió mediante los requisitos que se tenía por parte de la institución auspiciante.

Las imágenes que se visualizarán a continuación son extraídas del simulador de IBM Worklight Studio pero representa un dispositivo Samsung Galaxy S2.



Ilustración 48 Pantalla de Inicio de la aplicación

Fuente: Propia

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”



Ilustración 49 Pantalla de Logeo a la aplicación

Fuente: Propia



Ilustración 50 Pantalla de menú del Usuario

Fuente: Propia

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”



Ilustración 51 Pantalla de lista de materias para el usuario

Fuente: Propia



Ilustración 52 Pantalla de menú para consultar opciones académicas

Fuente: Propia

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

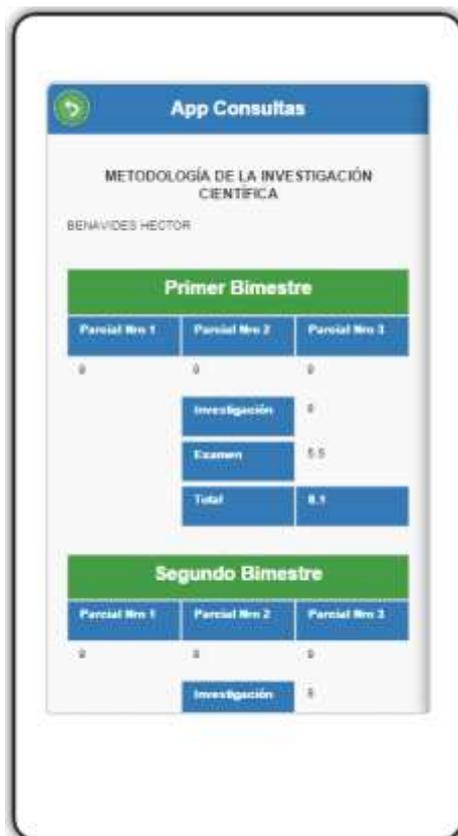


Ilustración 53 Pantallas de Notas del usuario

Fuente: Propia



Ilustración 54 Pantallas de Sílabos

Fuente: Propia

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”



Ilustración 55 Pantallas de Búsqueda Bibliográfica

Fuente: Propia



Ilustración 56 Pantalla de Información de Búsqueda de Bibliografía

Fuente: Propia

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- El desarrollo de aplicaciones móviles en estos tiempos ha crecido con el auge de dispositivos móviles que las personas poseen, a más personas comprenden dispositivos más aplicaciones se desarrollarán.
- Las herramientas de desarrollo de aplicaciones móviles cada vez están creciendo más, añadiendo más componentes y funciones para los desarrolladores.
En la actualidad existen herramientas de pago o gratuitas dependiendo la amplitud del proyecto o aplicación a desarrollar.
- IBM Worklight Studio es un software de desarrollo para aplicaciones móviles que poseen muchas funciones y herramientas de desarrollo por lo que sería buena estudiarla para mejorar los conocimientos en los estudiantes de la Universidad ya que en futuro podrían necesitar de este tipo de herramientas para su desempeño laboral en caso de trabajar en grandes empresas.
- IBM Worklight Studio ayudó que la aplicación se desarrolle muy ágilmente, ya que posee un adaptador que agiliza la obtención de datos por lo que se conectó directamente con las tablas sin ningún problema, como esta herramienta es de desarrollo multiplataforma ayudó a que el desarrollo para los tres sistemas operativos sea rápida y compatible para diferentes dispositivos por último gracias a la configuración de skin se logró adaptar a diferentes tamaños de dispositivos.

5.2. RECOMENDACIONES

- Promover a los estudiantes a la creación de más Trabajos de Grado investigativos, enfocados al desarrollo en plataformas de desarrollo móvil.
- Dentro de la asignatura de computación móvil se recomienda estudiar más sobre herramientas para desarrollo de aplicaciones móviles multiplataforma.
- Si se desea aprender más sobre el desarrollo multiplataforma la herramienta IBM Worklight Studio sería ideal para el aprendizaje y orientación hacia estas herramientas.

- Las tablas de la base de datos para desarrollar una aplicación o software de diferente tipo deben de estar documentadas para agilizar el desarrollo, se debe proporcionar a tiempo todos los recursos para mejorar el diseño de la aplicación.

5.3. GLOSARIO DE TÉRMINOS

- **IBM:** International Business Machines, es una empresa norteamericana que desarrolla software para creación de software y hardware de diferentes tipos
- **HTML5:** es la quinta versión de HTML, que es un lenguaje de marcado de etiquetas, con el que se puede desarrollar aplicaciones web. La quinta versión posee nuevas y funcionales herramientas y etiquetas que ayudan a generar rápidamente el código.
- **SO:** abreviación de Sistema Operativo
- **IBM Worklight Studio:** es una de las herramientas que posee IBM Worklight, pero esta es un framework en el que se puede crear software dirigido para aplicaciones móviles a nivel multiplataforma.
- **SCRUM:** es una metodología de desarrollo ágil de software, en el que interesa plasmar los requerimientos del usuario de manera rápida.
- **Android:** es un sistema operativo móvil creado por Google, su desarrollo es completamente libre.
- **iOS:** iPhone Operating System, es un sistema operativo móvil creado por Apple.
- **Windows Phone:** es un sistema operativo móvil desarrollado por Microsoft, como sucesor de la plataforma Windows Mobile.
- **IDE:** es un entorno de desarrollo integrado, donde se puede desarrollar programas.
- **Plug-in:** es un complemento de otra aplicación cuyo objetivo es aportar funciones nuevas a la aplicación.
- **Eclipse:** es un IDE de programación para lenguajes como java, c++, permite realizar proyectos para escritorio, web y móvil. Es libre de paga.
- **API's:** Application Programming Interface, es el conjunto de funciones y procedimientos, que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

- **SQL:** es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.
- **HTTP:** (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.
- **REST:** es una técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.
- **SOAP:** (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.
- **JMS:** Java Message Service, es un estándar de mensajería que permite a los componentes de aplicaciones basados en la plataforma Java2 crear, enviar, recibir y leer mensajes. También hace posible la comunicación confiable de manera síncrona y asíncrona.

5.4. REFERENCIAS BIBLIOGRÁFICAS

Adobe. (2014). *Adobe*. Obtenido de Adobe: <http://helpx.adobe.com/es/dreamweaver/using/web-applications.html>

Alto Nivel. (2013). *Alto Nivel*. Obtenido de Alto Nivel: <http://www.altonivel.com.mx/sistemas-operativos-para-moviles.html>

Castaño. (2014). *Sistemas Operativos a través del tiempo*.

Castro, J., & Cucker, F. (s.f.). *Curso de Programación*. McGraw-Hill.

Dan, T. s. (2014). *Sistema Operativo IOS*. Obtenido de shideshare: <http://es.shideshare.net/Ten shi Dan/sistema-operativo-ios>

Delgado, M. (2013). *Aplicaciones híbridas: presente y futuro del desarrollo Móvil*. Madrid: atSistemas. Obtenido de http://atsistemas.com/wp-content/uploads/2013/04/20130429_aplicaciones_hibridas.pdf

Doncel, M. (2010). *STARTCAPPS*. Obtenido de STARTCAPPS: <http://www.startcapps.com/blog/que-es-una-app-nativa/>

Estudio de la herramienta IBM Worklight Studio para el desarrollo de aplicaciones móviles multiplataforma con aplicativo de consulta de notas, libros y sílabos en el Instituto Tecnológico Superior “José Chiriboga Grijalva”

- Galeano, D. (05 de Octubre de 2012). *desarrolloWeb.com*. Obtenido de Introducción al framework PhoneGap: <http://www.desarrolloweb.com/articulos/inro-framework-phonegap.html>
- IBM. (2013). *IBM Knowledge Center*. Obtenido de IBM Knowledge Center: http://www-01.ibm.com/support/knowledgecenter/SSZH4A_6.2.0/com.ibm.worklight.dev.doc/devref/c_jsonstore_overview.html?lang=en
- IBM Corporation. (2013). IBM Worklight V5.0.6. En I. Corporation, *IBM Worklight V5.0.6* (pág. 12). New York: Software Group.
- IBM Worklight. (2014). *IBM*. Obtenido de IBM MobileFirst Platform Foundation: <http://www-03.ibm.com/software/products/es/mobilefirstfoundation>
- J.Castellanos. (2012). *Ventajas del XAML*. Obtenido de Ventajas del XAML: <http://e-archivo.uc3m.es/bitstream/handle/10016/17873/Memoria%20PFC,%20F.J.Castellanos.pdf?sequence=1>
- La revista informática. (2009). La revista informática. *La revista informática*, 50.
- LanceTalent. (20 de Febrero de 2014). *LanceTalent*. Obtenido de LanceTalent: <http://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/>
- Leandro, A. (2013). *DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA*. Obtenido de DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA: <http://www.alegsa.com.ar/Dic/aplicacion%20web.php>
- Pimienta, P. (17 de Abril de 2014). *de ideaAapp*. Obtenido de ¿Sabes qué es Phonegap y Phonegap Build?: <http://deideaaapp.org/sabes-que-es-phonegap-y-phonegap-build/>
- Scoello12. (2012). *Ventajas y Desventajas de Android*. Obtenido de Ventajas y Desventajas de Android: <http://wordpress.com/ventajas-desventajas/>
- UDE. (2013). *Sabia TIC*. Obtenido de Sabia TIC: Sabia.tic.ude.es
- Universidad de Nebrija. (2012). *Lenguaje de programación Java*.
- Windows. (2013). *Windows Phone*. Obtenido de Windows Phone: <http://www.microsoft.com>