



UNIVERSIDAD UTE

**FACULTAD DE CIENCIAS DE LA INGENIERÍA E
INDUSTRIAS**

**INGENIERÍA EN INFORMÁTICA Y CIENCIAS DE LA
COMPUTACIÓN**

**“DISEÑO DE UNA APLICACIÓN MÓVIL PARA
NOTIFICACIÓN DE EVENTOS DE UNA PLATAFORMA
MOOC”**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN INFORMÁTICA Y CIENCIAS DE LA
COMPUTACIÓN**

ANTONIO ENRIQUE IGLESIAS MESTANZA

ING. BOLÍVAR JÁCOME

© Universidad Tecnológica Equinoccial. 2020
Reservados todos los derechos de reproducción

FORMULARIO DE REGISTRO BIBLIOGRÁFICO TRABAJO DE TITULACIÓN

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	1722904388
APELLIDO Y NOMBRES:	IGLESIAS MESTANZA ANTONIO ENRIQUE
DIRECCIÓN:	CALLE E3 N74-266 Y JOAQUÍN MANCHENO
EMAIL:	imae103624@ute.edu.ec
TELÉFONO FIJO:	02-3500600
TELÉFONO MOVIL:	0984346280

DATOS DE LA OBRA	
TÍTULO:	“DISEÑO DE UNA APLICACIÓN MÓVIL PARA NOTIFICACIÓN DE EVENTOS DE UNA PLATAFORMA MOOC”
AUTOR O AUTORES:	IGLESIAS MESTANZA ANTONIO ENRIQUE
FECHA DE ENTREGA DEL PROYECTO DE TITULACIÓN:	18 DE AGOSTO DEL 2018
DIRECTOR DEL PROYECTO DE TITULACIÓN:	ING. BOLÍVAR JACOME
PROGRAMA	PREGRADO <input checked="" type="checkbox"/> PROGRADO <input type="checkbox"/>
TÍTULO POR EL QUE OPTA:	Ingeniero en Informática y Ciencias de la Computación
RESUMEN:	El presente trabajo de titulación tiene como objetivo principal diseñar una aplicación móvil para la notificación de eventos de una plataforma MOOC. Donde su metodología de desarrollo se describe paso a paso, empezando con la creación y desarrollo de un curso MOOC acerca del Internet de las Cosas (IoT), además de una aplicación que notifique al usuario actividades y fechas importantes que se debe cumplir a lo

	<p>largo del mismo, se ha solicitado que este proyecto esté vinculado al proyecto CEDIA: CIUDADES INTELIGENTES, por lo cual se ha visto necesario que el curso sea acerca de los fundamentos de IoT, lo cual se integrara a proyectos futuros en relación a cursos que la universidad desee impartir. En los resultados se observa una aplicación para el sistema operativo Android, totalmente funcional, la cual se conecta con una base de datos de tipo MySQL, que hace consultas sobre las actividades más importantes del curso “FUNDAMENTOS DEL INTERNET DE LAS COSAS (IOT)”, desarrollado en la plataforma MOOC llamada OpenEDX, esta aplicación notifica al usuario por medio de Notificaciones Push ejecutadas en segundo plano, sobre estas actividades que recordaran al usuario el desarrollo de estas en el curso anteriormente mencionado. Se concluye que la implementación de un curso sobre fundamentos de IoT en una plataforma MOOC en conjunto con el desarrollo de una aplicación móvil conectada a la base de datos del mismo, permite la notificación de actividades y fechas importantes del mismo, por medio de notificaciones Push que se ejecutaran en actividades de segundo plano en el Sistema Operativo Android.</p>
PALABRAS CLAVES:	IoT, MOOC, Mobile-D, Android, Aplicación, CEDIA.
ABSTRACT:	The main objective of this degree work is to design a mobile application for the notification of events of a MOOC platform. Where its development methodology is described step by step, starting with the creation and development of a MOOC course about the Internet of Things (IoT), as well as an application that notifies the user of important activities and dates that must be met throughout of the same. It has

	<p>been requested that this project be linked to the CEDIA: INTELLIGENT CITIES project, which is why it has been necessary for the course to be about the fundamentals of IoT, which will be integrated into future projects in relation to courses that the university wishes to teach. The results show an application for the Android operating system, fully functional, which connects to a MySQL-type database, which makes inquiries about the most important activities of the course "FUNDAMENTALS OF THE INTERNET OF THINGS (IOT)" , developed on the MOOC platform called OpenEDX, this application notifies the user by means of Push Notifications executed in the background, about these activities that will remind the user of the development of these in the aforementioned course. It is concluded that the implementation of a course on fundamentals of IoT in a MOOC platform in conjunction with the development of a mobile application connected to its database, allows the notification of activities and important dates thereof, through Push notifications that will run in background activities on the Android Operating System.</p>
KEYWORDS	IoT, MOOC, Mobile-D, Android, Application, CEDIA.

Se autoriza la publicación de este Proyecto de Titulación en el Repositorio Digital de la Institución.

A handwritten signature in blue ink, consisting of several overlapping loops and lines, positioned above a dashed line.

F.-----

IGLESIAS MESTANZA ANTONIO ENRIQUE

1722904388

DECLARACIÓN Y AUTORIZACIÓN

Yo, **ANTONIO ENRIQUE IGLESIAS MESTANZA**, CI 1722904388 autor/a del proyecto titulado **DISEÑO DE UNA APLICACIÓN MÓVIL PARA NOTIFICACIÓN DE EVENTOS DE UNA PLATAFORMA MOOC** previo a la obtención del título de Ingeniero en Informática y Ciencias de la Computación en la Universidad Tecnológica Equinoccial.

1. Declaro tener pleno conocimiento de la obligación que tienen las Instituciones de Educación Superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Autorizo a la BIBLIOTECA de la Universidad UTE tener una copia del referido trabajo de graduación con el propósito de generar un Repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Quito, 18 de Agosto del 2020



F.-----

IGLESIAS MESTANZA ANTONIO ENRIQUE

1722904388

CERTIFICACIÓN DEL TUTOR

Certifico que el presente trabajo que lleva por título “DISEÑO DE UNA APLICACIÓN MÓVIL PARA NOTIFICACIÓN DE EVENTOS DE UNA PLATAFORMA MOOC”, que, para aspirar al título de Ingeniero en Informática y Ciencias de la Computación fue desarrollado por Iglesias Mestanza Antonio Enrique, bajo mi dirección y supervisión, en la Facultad de Ciencias de la Ingeniería; y cumple con las condiciones requeridas por el reglamento de Trabajos de Titulación artículos 19, 27, 28.



F.-----

Ing. Bolívar Jácome

DIRECTOR DEL TRABAJO

DECLARACIÓN JURAMENTADA DEL AUTOR

Yo ANTONIO ENRIQUE IGLESIAS MESTANZA, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Tecnológica Equinoccial puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

A handwritten signature in blue ink, appearing to read 'ANTONIO ENRIQUE IGLESIAS MESTANZA', is written over a horizontal line.

F.-----

IGLESIAS MESTANZA ANTONIO ENRIQUE

1722904388

DEDICATORIA

El presente trabajo es dedicado a mi familia, mi madre Teresa, que siempre, a más de enseñarme a vivir, con su amor, cariño incondicional y su guía constante siempre ha sido un apoyo fundamental, no solo para mi formación, sino para lograr ser una buena persona, y entender que en la vida, contaras con poca gente que esté dispuesta a darlo todo por ti como ella lo ha hecho conmigo y con mi hermana

A mi padre Antonio, que nunca le importo despertar temprano, acostarse tarde, y viajar lejos solo por darnos un plato que comer a toda nuestra familia, como también enseñarme que hay cosas que, si logras por ti mismo, no importa que no haya grandes reconocimientos y bellos laureles, lo que cuenta es que pusiste todo tu empeño en ellas.

A mi hermana Abigail, me enseñó que, detrás de todo esfuerzo, siempre hay una recompensa, pero que lograr tus objetivos no es tarea fácil, y siempre debe haber un compromiso ante todo para merecer ganar las batallas que te da la vida, los dos aprendemos uno del otro, pero cada día con ella es todo un aprendizaje constante, divertido y de mucho cariño.

Y finalmente, este trabajo es dedicado a Wendy, que me enseñó que, cuando parece que la vida es oscura y sin sentido, siempre hay un rayo de luz que llenara de alegría tus días. Tú eres ese rayo de luz cuando me brindas tu sonrisa, y pase lo que pase en el futuro, siempre serás el amor de mi vida, Te amo.

AGRADECIMIENTOS

Agradezco a mi madre Teresa, a mi padre Antonio y a mi hermana Abigail, ya que junto con su cariño e instrucción, he logrado ser una persona instruida, educada y el presente trabajo es un pequeño testimonio de esto, gracias por estar en mi vida, además de siempre darme su amor y cariño incondicional, los amo y gran parte de mi vida son ustedes, espero poder ser mejor hijo y hermano, y perdón si los decepcione o herí a veces.

También agradezco a mi tutor, el Ing. Bolívar Jácome, además de a los Ingenieros Ciro Saguay, Patricio Armas, Rodrigo Proaño y Oswaldo Moscoso, ya que la concepción de este trabajo de titulación fue en parte, logro de todos ellos.

A la vez agradezco al Ingeniero Paul García y al Ingeniero Diego Ortiz, ya que sin la asesoría de ellos varios aspectos de esta tesis no hubieran sido posibles

Y finalmente agradezco a Wendy, sin la cual no hubiera tenido las fuerzas de despertar cada día, y completar este trabajo, espero que la vida me alcance para demostrarle cuanto le agradezco, no solo por darme la fuerza de completar esto, sino también por estar en mi vida y ser la razón de mi alegría cuando todo parecía tristeza y soledad.

ÍNDICE DE CONTENIDOS

	PÁGINA
RESUMEN	1
ABSTRACT	2
1. INTRODUCCIÓN	4
2. METODOLOGÍA	14
2.1. Metodología Mobile-D	14
2.1.1. Fase 1: Exploración	14
2.1.2. Fase 2: Inicialización del Producto.....	14
2.1.3. Fase 3: Producción	15
2.1.4. Fase 4: Estabilización	15
2.1.5. Fase 5: Pruebas.....	15
2.2. Metodología XMOOC	16
2.2.1. Fase 1: Tipo de curso y plataforma.....	16
2.2.2. Fase 2: Diseño del curso	16
2.2.3. Fase 3: Estructura, contenido y secuencias didácticas.....	16
2.2.4. Fase 4: Evaluación y Canales de comunicación y nivel interacción.....	16
3. RESULTADOS Y DISCUSIÓN	19
3.1. Aplicación Móvil.....	19
3.1.1. Fase 1: Exploración	19
3.1.2. Fase 2: Inicialización Del Producto	21
3.1.3. Fase 3: Producción	24
3.1.4. Fase 4: Estabilización	30
3.1.5. Fase 5: Pruebas.....	33
3.2. Metodología XMOOC	34
3.2.1. Fase 1: Tipo de curso y plataforma.....	34
3.2.2. Fase 2: Diseño Del Curso	38
3.2.3. Fase 3: Estructura, contenido y secuencias didácticas.....	38
3.2.4. Fase 4: Evaluaciones y Canales de Comunicación	47
4. CONCLUSIONES Y RECOMENDACIONES	55
4.1. CONCLUSIONES.....	55

4.2. RECOMENDACIONES	55
5. BIBLIOGRAFÍA	58
6. ANEXOS.....	62

ÍNDICE DE FIGURAS

	PÁGINA
Figura 1. Diagrama de casos de Uso 1.	21
Figura 2. Prototipo Interfaz de Aplicación.....	26
Figura 3. Algoritmo método DownloadUrl.....	26
Figura 4. Algoritmo ConsultarDatos	27
Figura 5. Algoritmo Método OnCreate.....	27
Figura 6. Algoritmo Método OnStartCommand.	28
Figura 7. Algoritmo Método OnDestroy.	28
Figura 8. Diagrama de flujo de procesamiento de notificaciones.....	29
Figura 9. Diagrama de flujo de ingreso al sistema.....	29
Figura 10. Diagrama de flujo de ingreso a ingreso al método Receiver Call.....	30
Figura 11. Código conexión de Aplicación a Base de datos MySQL.	31
Figura 12. Código consulta de Aplicación a tablas de Base de datos MySQL.....	31
Figura 13. Consulta de aplicación a Base de datos por medio de código PHP.	31
Figura 14. Algoritmo Creación de Notificaciones.....	32
Figura 15. Gráfico comparativo Plataformas MOOC	35
Figura 16. Comando Ejecución Ironwood.....	35
Figura 17. Pantalla de Instalación Ironwood.....	36
Figura 18. Pantalla de Instalación Ironwood.....	36
Figura 19. Credenciales Autenticación Ironwood.	36
Figura 20. Hostname Ironwood.	37
Figura 21. Plataforma OpenEDX Ironwood.	37
Figura 22. Servidores Plataforma OpenEDX Ironwood.	37
Figura 23. Estructura Curso “Fundamentos Del Internet De Las Cosas (IoT)”	40
Figura 24. Recurso Didáctico en plataforma OpenEDX.....	41
Figura 25. Trasccripción Videos en plataforma OpenEDX.	41
Figura 26. Nueva Unidad de Curso en plataforma OpenEDX.....	42
Figura 27. Componente de Video en Unidad de Curso en plataforma OpenEDX...42	42
Figura 28. Configuraciones de Componente de Video en Unidad de Curso en plataforma OpenEDX.....	42
Figura 29. Publicación Presentaciones en la Web Power Point	43
Figura 30. Incorporación de Presentación en formato HTML	43
Figura 31. Plantillas HTML.....	44
Figura 32. Código HTML Presentación Power Point en OpenEDX.....	44
Figura 33. Presentación Power Point en OpenEDX.	44
Figura 34. Publicación Presentaciones en la Web Prezi	45
Figura 35. Incorporación de Prezi en formato HTML	45
Figura 36. Plantillas HTML.....	46
Figura 37. Código HTML Presentación Prezi en OpenEDX	46
Figura 38. Presentación Prezi en OpenEDX	47
Figura 39. Modelo Preguntas Curso MOOC.....	48
Figura 40. Creación de Directorio de Migración EDX.	49
Figura 41. Migración Base de datos MySQL en EDX.....	49
Figura 42. Migración Base de datos Mongo en EDX.....	49

Figura 43. Migración Archivos de datos en EDX.....	49
Figura 44. Migración Archivos .Json en EDX.	49
Figura 45. Migración Base de datos Mongo en EDX.....	49
Figura 46. Compresión Datos Plataforma EDX.	49
Figura 47. Software FilleZilla.....	50
Figura 48. Extracción de datos plataforma EDX.....	50
Figura 49. Migración Base de datos MySQL plataforma EDX.	50
Figura 50. Migración Base de datos MySQL plataforma EDX.....	50
Figura 51. Migración de datos plataforma EDX.....	50
Figura 52. Migración LMS plataforma EDX.	51
Figura 53. Actualización LMS plataforma EDX.....	51
Figura 54. Migración datos Queue plataforma EDX.	51
Figura 55. Reinicio Servidores plataforma EDX.....	51
Figura 56. Google Open Plataform OpenEDX.....	51
Figura 57. Cursos Plataforma EDX.....	51
Figura 58. Curso “Fundamentos del Internet de las Cosas”	52
Figura 59. Comando Obtención Script Base de Datos MySQL EDX	52
Figura 60. Script Base de Datos MySQL EDX	52
Figura 61. Ejecutor de Script Base de Datos MySQL.....	53
Figura 62. Tabla “course_overview_courseoverview” de Base de Datos MySQL OpenEDXOpenEDX	53

ÍNDICE DE TABLAS

PÁGINA

Tabla 1. Diferencia entre MOOCs y cursos en línea.....	6
Tabla 2. Diferencias entre Metodologías cMOOC y xMOOC.....	8
Tabla 3. Diferencias entre las Metodologías de Cascada, Desarrollo Rápido, Desarrollo Ágil y Mobile-D.....	12
Tabla 4. Fase 1 Exploración.....	19
Tabla 5. Caso de Uso No 1.....	21
Tabla 6. Fase 2 Inicialización del Producto.....	22
Tabla 7. Fase 3 Producción.....	24
Tabla 8. Fase 4 Estabilización.....	30
Tabla 9. Pruebas Funcionales del Sistema.....	33
Tabla 10. Fase 1: Tipo de curso y plataforma.....	34
Tabla 11. Cuadro comparativo Plataformas MOOC.....	34
Tabla 12. Fase 2: Diseño del curso.....	38
Tabla 13. Fase 3 Estructura, contenido y secuencias didácticas.....	39
Tabla 14. Fase 4 Evaluación.....	47

ÍNDICE DE ANEXOS

	PÁGINA
Anexo 1. Código MainActivity	62
Anexo 2. Código MyService	66
Anexo 3. Código ReceiverCall	72
Anexo 4. Código SplashActivity	73
Anexo 5. Código Activity_Main	74
Anexo 6. Código activity_splash_activity	76
Anexo 7. Splash Screen Aplicación Notificaciones Edx	77
Anexo 8. Aplicación Notificaciones Edx	78
Anexo 9. Plataforma OpenEdx Studio Curso "FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)"	79
Anexo 10. Plataforma OpenEdx LMS Curso "FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)"	80

RESUMEN

El presente trabajo de titulación tiene como objetivo principal diseñar una aplicación móvil para la notificación de eventos de una plataforma MOOC. Donde su metodología de desarrollo se describe paso a paso, empezando con la creación y desarrollo de un curso MOOC acerca del Internet de las Cosas (IoT), además de una aplicación que notifique al usuario actividades y fechas importantes que se debe cumplir a lo largo del mismo, se ha solicitado que este proyecto esté vinculado al proyecto CEDIA: CIUDADES INTELIGENTES, por lo cual se ha visto necesario que el curso sea acerca de los fundamentos de IoT, lo cual se integrara a proyectos futuros en relación a cursos que la universidad desee impartir. En los resultados se observa una aplicación para el sistema operativo Android, totalmente funcional, la cual se conecta con una base de datos de tipo MySQL, que hace consultas sobre las actividades más importantes del curso “FUNDAMENTOS DEL INTERNET DE LAS COSAS (IOT)”, desarrollado en la plataforma MOOC llamada OpenEDX, esta aplicación notifica al usuario por medio de Notificaciones Push ejecutadas en segundo plano, sobre estas actividades que recordaran al usuario el desarrollo de estas en el curso anteriormente mencionado. Se concluye que la implementación de un curso sobre fundamentos de IoT en una plataforma MOOC en conjunto con el desarrollo de una aplicación móvil conectada a la base de datos del mismo, permite la notificación de actividades y fechas importantes del mismo, por medio de notificaciones Push que se ejecutaran en actividades de segundo plano en el Sistema Operativo Android.

Palabras Clave: IoT, MOOC, Mobile-D, Android, Aplicación, CEDIA.

ABSTRACT

The main objective of this degree work is to design a mobile application for the notification of events of a MOOC platform. Where its development methodology is described step by step, starting with the creation and development of a MOOC course about the Internet of Things (IoT), as well as an application that notifies the user of important activities and dates that must be met throughout of the same. It has been requested that this project be linked to the CEDIA: INTELLIGENT CITIES project, which is why it has been necessary for the course to be about the fundamentals of IoT, which will be integrated into future projects in relation to courses that the university wishes to teach. The results show an application for the Android operating system, fully functional, which connects to a MySQL-type database, which makes inquiries about the most important activities of the course "FUNDAMENTALS OF THE INTERNET OF THINGS (IOT)" , developed on the MOOC platform called OpenEDX, this application notifies the user by means of Push Notifications executed in the background, about these activities that will remind the user of the development of these in the aforementioned course. It is concluded that the implementation of a course on fundamentals of IoT in a MOOC platform in conjunction with the development of a mobile application connected to its database, allows the notification of activities and important dates thereof, through Push notifications that will run in background activities on the Android Operating System.

Keywords: IoT, MOOC, Mobile-D, Android, Application, CEDIA.

1. INTRODUCCIÓN

1. INTRODUCCIÓN

El mundo actual está caracterizado por el crecimiento exponencial desde el enfoque social, económico y tecnológico. Vivimos en un entorno altamente dependiente de la tecnología, una sociedad digital o digitalizada, y esto está trayendo y traerá consecuencias en muchos ámbitos de nuestras vidas a una velocidad tal que las hace difícil de prever. La comunidad encargada de la ciencia pide impulsar, promocionar y reactivar el desarrollo y crecimiento económico de los países, empleando la tecnología como un instrumento eficaz puesto al servicio de la alfabetización de la población y la democratización del conocimiento (García G., 2018). Desde la perspectiva de la pedagógica inclusiva y colaborativa, es pertinente el aprovechamiento de la tecnología para la creación de espacios de aprendizaje, mediante comunidades virtuales para la construcción social del conocimiento. En nuestro medio, los cursos MOOC aprovechan los recursos de la web y dan respuesta a las necesidades de formación de nuestra sociedad, brindando entornos de aprendizaje no formales que brinden servicio a toda la sociedad desde una perspectiva democrática, colaborando en la disminución de la grieta social y digital.

Para el desarrollo de una aplicación móvil, es necesario usar una metodología que permita manejar de forma ordenada los procesos de construcción de la aplicación para que ayude a mejorar ciertas funciones que al usuario le sea difícil realizar.

La enseñanza se define como un proceso instructivo orientado a educar a terceros, facilitando su aprendizaje (Álvarez & Cardona 2011). La educación ha sido a lo largo de los años una herramienta de transmisión de conocimiento efectiva, y en la actualidad, está se ha visto replanteada desde cursos de asistencia presencial, a cursos a distancia, sin necesidad de recurrir a centros educativos. La demanda de estos cursos se ha incrementado gracias a las necesidades de dar una respuesta más veloz y efectiva de educación, cumpliendo las actividades que un estudiante realiza diariamente sin la necesidad de salir de casa, ajustándose a sus horarios, los docentes, por otra parte, evalúan el progreso de sus estudiantes de manera permanente, usando distintas herramientas, tales como pruebas, actividades y talleres. En un mercado cada vez más competitivo, los cursos MOOC facilitan la educación en línea.

Para entender la diferencia entre un curso online con un curso MOOC es menester analizar sus siglas cuyo significado es "Massive Online Open Courses" que en español significa Cursos online abiertos y masivos, planteada para modalidades a distancia. No tiene una limitación de participantes debido a su singularidad de fácil adaptación e interactividad con

el usuario, además de ser accesible por medio del internet, sin limitación o discriminación alguna. (Universitat Autònoma de Barcelona, 2013)

Seaton. 2018, en su artículo “Studying Learning in the Worldwide Classroom Research into edX’s First MOOC”, muestra estadísticas de nuevos usuarios en el ámbito de las plataformas MOOC. Según la MIT, en estos cursos se inscribieron estudiantes de 194 países; y entre los cinco primeros estuvieron Estados Unidos, India, Reino Unido, Colombia y España, de esta manera queda demostrado que el uso de plataformas MOOC además de facilitar el autoaprendizaje, elimina fronteras.

El término MOOC fue dado forma por Dave Cormier (2008) haciendo referencia al curso “Conectivismo y Aprendizaje Conectado”, impartido por en la Universidad de Manitoba, por los canadienses Stephen Downes y George Siemens. Desde entonces se han ejecutado habitualmente distintos MOOCs, enfocados principalmente en tecnologías emergentes de tipo educacional y en las hipótesis de aprendizaje. (Universidad Carlos III de Madrid, 2020)

En el 2017 se denota un incremento significativo de estudiantes que acceden a estas plataformas, las inscripciones a estos cursos desde su creación en el 2011, ascendieron a 23 millones de estudiantes inscritos, con un total de 81 millones de estudiantes en línea, varios de ellos registrados en estas plataformas (Shah, 2018):

- Coursera - 30 millones
- edX - 14 millones
- XuetangX - 9.3 millones
- Udacity - 8 millones
- FutureLearn - 7.1 millones

A partir de esto, el interés a estos cursos se ha incrementado dramáticamente, explicándose así, el interés de institutos de educación superior en implementarlos, además que, logrando disminuir significativamente sus costos operativos alcanzando un mayor margen de utilidad, ven en esta modalidad una gran alternativa para educar a más estudiantes usando menos recursos (Universidad de Alicante, 2012). Sin embargo, es importante tener claro las diferencias que tiene un curso MOOC con un Curso Online simple. En la Tabla 1 se define ámbitos para la identificación de un curso MOOC como de un curso en línea:

Tabla 1. Diferencia entre MOOCs y cursos en línea.

MOOC	Curso en línea
Entorno abierto.	Entorno cerrado.
Se parte de un modelado tecnológico que ayuda la difusión de la actividad de los involucrados por medio de la utilización de estas plataformas.	Se despliega en la plataforma de enseñanza con funciones y mecanismos muy sintetizados y contruidos para la relación directa con el tutor.
Fomenta el protagonismo del alumnado.	Acceso por pago previo de matrícula.
Número de participantes ilimitado.	Número de participantes limitado.
Gran cantidad de instrumentos de comunicación, manejo de redes sociales.	Comunicación por medio de email y foros de debate.
Énfasis en el proceso de aprendizaje más que en la evaluación y la acreditación.	Orientado para lograr la acreditación por medio de la evaluación.
Acceso y materiales gratuitos.	El estudiante debe responder mediante modelos de instrucción básicos.

(Universitat Autònoma de Barcelona, 2013)

Schamer, 2015 explica que en el perfeccionamiento de los MOOC, se adecuó el diseño de un escenario general, donde el papel del estudiante se lleva a cabo en una nueva área de instrucción en donde todos los niveles tiene un método predefinido. Existen distintos niveles de educación MOOC entre las cuales se encuentran:

- Educación de Espacio Uno a Uno: El estudiante genera contenido y soluciona problemas de acuerdo con lo predefinido por el educador.
- Educación de Uno a Muchos: El educador da una sesión a muchas personas a la vez.
- Educación de Muchos a Muchos: Se explica que el diálogo con el alumno es fundamental, se pone en consideración la transmisión de conocimientos de las partes involucradas.

El núcleo de la filosofía de los cursos MOOC subsiste en el desarrollo de código abierto aplicado a las plataformas disponibles, haciendo referencia a estas, SCI FLO PORTUGAL, 2014 señalan como principal escenario a OpenEDX, disponible desde junio de 2013, la misma que tiene un diseño pedagógico sencillo, esta es una de las características más sobresalientes considerando el desenvolvimiento de los estudiantes y procurando el cumplimiento de objetivos y metas señaladas por quienes la usan.

Carina S. González, 2016 indica que MOOC tiene un diseño muy novedoso, el cual incentiva al favorecimiento del aprendizaje y mitiga la deserción temprana de los cursos, a la vez que propicia la colaboración de todos los

involucrados y al desarrollo de técnicas de expansión en la planificación, diseño e incorporación de los MOOC en entornos de producción.

Carlos Castaño Garrido, 2015 define que los MOOC están determinados por el gran número de participantes y por la variedad de temas que se puede encontrar en las diversas plataformas de desarrollo. Sin embargo, estos factores no impiden que haya un alto porcentaje de deserción. Por ello, se propone un diseño mixto, cooperativo y basado en el desarrollo de las redes sociales que puede ser tomado como estrategia para el desarrollo del aprendizaje, y así mitigar el abandono de los cursos

Gracias a la facilidad de acceso a estos cursos, países como España han logrado mitigar el índice de deserción universitaria, sin embargo su éxito es de apenas un 10%, entre las causas más frecuentes están las obligaciones laborales y/o familiares, olvido de fechas importantes de eventos de la plataforma, una temática poco amigable para el usuario y contenidos nada interesantes que en lugar de capacitar, causen frustración y fatiga. (Vázquez, López, & Sarasola, 2013)

Se han determinado dos clases principales de metodologías MOOC las cuales son:

- XMOOC: Son MOOC dirigidos a usuarios con mayor experticia, donde el tutor agrega contenido que será encaminado a sus estudiantes. El profesor es la figura central del aprendizaje y su rol es crear y coordinar un plan de estudios estableciendo resultados desde el inicio.
- CMOOC: Está centrada en que la educación del alumno dependerá del uso que el mismo dé a las herramientas que le proporcione su profesor, no hay un plan de estudios formal o un sistema de gestión tradicional de aprendizaje, el cual se ve sustituido por módulos Web 2.0, en las que encontramos a blogs, wikis y plataformas de redes sociales (Sergio Luján Mora, 2013). En la Tabla 2 se encuentra aspectos que determinan las diferencias entre estos MOOCs.

Tabla 2. Diferencias entre Metodologías cMOOC y xMOOC.

Indicadores	Metodología xMOOC	Metodología cMOOC
Estrategias	Conectivista.	Conductista.
Similitud con	Entornos sociales y personales de aprendizaje.	Cursos <i>online</i> LMS.
Guía del aprendizaje	Aprendizaje colaborativo a partir de la interacción entre los participantes.	Se deriva de las actividades y/o contenidos.
Tipo de conocimiento	Conocimiento holístico.	Conocimiento empírico.
Evaluación	Creación de entornos individuales de aprendizaje.	Test. Entrega de tareas y talleres. Toma de pruebas en modalidad por pares.
Papel del docente	El docente tiene un perfil constructor de toda la comunidad.	El docente se convierte en el guía del aprendizaje.

(Cabero Almenara, Llorente Cejudo, & Vázquez Martínez, 2014)

Se han determinado 4 clases de participantes de cursos MOOCs los cuales son:

- Participantes pasivos: Son aquellos estudiantes que usan los materiales didácticos del curso, pero no interactúan en ninguna actividad de comunicación con los involucrados en el proceso de educación.
- Participantes activos: Hace referencia a los estudiantes que usan todos los materiales didácticos y realizan todas o muchas de las evaluaciones que se ofrecen dentro del curso.
- Participantes desconectados: Alumnos que realizan las evaluaciones al inicio del MOOC pero solo acceden a pocas clases o desaparecen por completo del curso.
- Participantes catadores o merodeadores: Estudiantes que exploran superficialmente el curso mediante el uso de algunos recursos didácticos (René Kizilcec, 2013).

Se han delimitado 4 roles principales de cursos MOOCs los cuales son:

- Narrativa: Las concepciones del docente sobre el tema de conocimiento deben estar al alcance del alumno, el profesor deberá lograr un acuerdo con los alumnos acerca de los objetivos de conocimiento.
- Reflexiva: El docente debe ayudar con el procedimiento a través del cual el alumno enlaza su entorno de educación con los objetivos de cada actividad y con los procedimientos finales de conocimiento.

- Adaptabilidad: El docente tiene la responsabilidad de establecer relaciones de sus propios conceptos con los del estudiante, para definir el enfoque de las tareas más adecuadas y propiciar el diálogo continuo entre alumno y maestro.
- Interactividad: El profesor debe transmitir un entorno de educación en el cual, el alumno pueda generar y dar respuestas a las actividades orientadas a la obtención de un objetivo (Berrocoso, 2014).

El internet de las cosas es la noción de una interinflexión digital de cuerpos cotidianos con internet. Esta tecnología constituye una red de objetos físicos que utiliza sensores y API's (Interfaz de programación de aplicaciones, bibliotecas que contienen funciones y procedimientos almacenados) para conectar e intercambiar datos a través de internet, con el objetivo de transformar los datos recibidos y que sean procesados a información (Zanjireh & Larijani, 2015)

En resumen, es el conjunto de lo que conocemos como internet, dando prioridad a la conexión con los objetos que nos rodean y con los individuos que los manejan. Esta tecnología da una alta prioridad a los datos que se pueden recolectar de todos los objetos que se encuentran en la interconexión, y no prioriza las acciones del usuario sino a la recopilación de datos otorgados por el objeto. (Evans, 2011)

Si los dispositivos eléctricos de uso diario tuvieran integradas identificaciones de algún tipo, podrían ser localizados de manera más simple y manejadas a alto nivel por otros mecanismos de la misma manera que si lo hiciera el usuario físicamente, logrando que sus interacciones físicas sean menos tediosas para los beneficiarios y los puedan manejar desde cualquier lugar de manera remota. (Magrassi, Panarella, Deighton, & Johnson, 2011)

Las aplicaciones para terminales conectadas a internet son diversas. Varias categorías han sido planteadas a través del tiempo. Greenough, 2015 ha propuesto separar en tres diferentes categorías que han sido aceptadas por la comunidad dedicada al estudio de IoT y son:

- Categoría enfocada a consumidores
- Categoría enfocada al sector empresarial
- Categoría enfocada a la infraestructura.

HM Treasury and The Rt Hon George Osborne, 2015 Propone que el Internet de las Cosas es el próximo escaño en la revolución de la información, relacionando a la interacción de los dispositivos con todos los servicios básicos que nos rodean, tanto de transporte, salud e inclusive considerando el uso de electrodomésticos de una familia promedio a través de dispositivos móviles.

Un porcentaje creciente de los dispositivos IoT son generados para el uso permanente de los usuarios, se incluye entre estos a los usuarios del sector automovilístico, conexiones digitales, ocio y entretenimiento, monitorización del domicilio, prendas inteligentes a partir de tecnología, monitoreo de estado de salud y equipos electrónicos para el hogar tales como lavadoras, secadoras, aspiradoras robóticas, purificadores de aire, hornos, refrigeradores que usan Wi-Fi para permitir el acompañamiento de manera remota en sus mecanismos (Trakin Tech Network, 2017).

A pesar de que existan artefactos antiguos, con poca capacidad en su un procesador, memoria y luz eléctrica, IoT daría la posibilidad de interconectarlos, demostrando así, que esta tecnología puede estar aplicada en casi cualquier ámbito. Estos sistemas podrían encomendarse a recopilar información en distintos ambientes: desde medios ecológicos hasta en el sector de construcción, por lo que podrían ser utilizados para control de la naturaleza y arquitectura urbana. (Mitchell, Villa, Stewart-Weeks, & Lange, 2013)

Algunas aplicaciones de uso común han estado bajo la lupa por su falta de respuestas a otros dispositivos y también por falta de consistencia. Varias empresas han sido cuestionadas por su inmersión temprana a IoT, desarrollando así dispositivos de dudosa reputación, además no logra captar el interés del consumidor, además de no fijar ni incorporar estándares de y adecuada de sus dispositivos. (PORUP, 2016)

Es un sistema operativo global enfocado para dispositivos móviles, con el objetivo de que estos estén contenidos en una plataforma abierta. La firma Android fue desarrollada por una empresa homónima, que, luego de ser comprada fue absorbida por Google en 2005, desde ese día, Google deposita gran cantidad de dinero y recursos de todo tipo para el desarrollo de este sistema operativo además de proyectos que varios desarrolladores puedan proponer para el mismo. (Burton, 2012)

El sistema operativo Android está actualmente desarrollado para pantallas de mayor resolución, cuenta con varias bibliotecas gráficas tanto 2D como 3D, soporte de OpenGL 2.0, y adaptación VGA para teléfonos tradicionales. Cuenta con un almacenamiento de corte SQLite, que funciona como base de datos liviana para almacenamiento de información, también dispone de alta conectividad y navegación web, Soporte de Java, la cual es fundamental a la hora del desarrollo de aplicaciones móviles, estas se ejecutan gracias a Dalvik, máquina virtual embebida en Android. (Google Developers, 2020)

Las aplicaciones en su estado original cuentan con cliente de email, SMS, calendarios, mapas, y están desarrolladas en lenguajes de programación Java. A la vez cuenta con un entorno de trabajo de aplicaciones, donde

algunos desarrolladores lo usan para tener total acceso a varias API's, que junto a varios frameworks, facilitan la reutilización de componentes y además pueden ser reemplazados por el usuario (Vanegas, 2012)

A la vez cuenta con un compilado de bibliotecas C/C++, que se exhiben a los desarrolladores en el entorno de labor de aplicaciones de Android, entre las principales librerías tenemos a System C Library (implementación de la biblioteca C estándar), bibliotecas de gráficos 3D, bibliotecas de medios además de una Base SQLite. Estos archivos son llamadas por varios componentes del sistema (Amaro, 2012)

Android depende casi en su totalidad de un kernel Linux. Gracias a su distribución de plataforma abierta, este Kernel provee de servicios básicos para todo el sistema operativo Android como gestión de procesos, gestión de almacenamiento y memoria, seguridad, modelo de controladores y pila de red. Este núcleo virtual también actúa como capa abstracta entre el dispositivo y el resto de la pila de las aplicaciones (Blanco, Camarero, Fumero, Warterski, & Rodríguez, 2009)

El desarrollo de aplicaciones Android es considerado un modelo de negocio bastante rentable, debido a que su desarrollo estratégico tiene muy en cuenta factores, herramientas y guías corporativas que han sido implementadas por varios expertos en tendencias de negocios de alto impacto a través del tiempo y que han convertido al mismo Android en un modelo de negocio como tal, y al ser altamente personalizable, Google lo emplea como ventaja para el perfeccionamiento de su sistema operativo (Android by Google, 2020)

Actualmente las aplicaciones de Android se desarrollan en lenguaje de programación Java, en conjunto con Android Software Development Kit también llamado Android SD. Estas aplicaciones, al estar construidas en el lenguaje de programación Java, no es necesario el aprendizaje de otro lenguaje, solo se requiere un kit de desarrollo de Software llamado SDK que contiene herramientas aplicadas en el desarrollo enfocado a software móvil (RDFI, 2008).

La demanda de aplicaciones de corte móvil ha visto un cambio radical con una expansión exponencial en nuestra época. Varias plataformas móviles siguen mejorando su desempeño y considerando la alta demanda de parte de los usuarios por lo que se ha visto la necesidad de ampliar el desempeño de estas aplicaciones. (Restivo, 2019)

Estas metodologías son un compilado de distintas técnicas, las cuales son una herencia de lo visto en prácticas comunes usadas en desarrollo de software, la cuales cuentan con pruebas que verifiquen el funcionamiento de las mismas. En la Tabla 3 se encuentra aspectos que determinan las

diferencias entre varias metodologías enfocadas al desarrollo móvil disponibles.

Tabla 3. Diferencias entre las Metodologías de Cascada, Desarrollo Rápido, Desarrollo Ágil y Mobile-D.

Indicadores	Cascada	Desarrollo Rápido	Desarrollo Ágil	Metodología cMOOC
Requisitos	Requisitos previamente definidos por el cliente	Obtención de prototipo funcional a partir de requisitos de cliente	Adaptabilidad a cambios de Requisitos	Planificación de conceptos básicos del Proyecto
Equipos de Desarrollo	Equipo de desarrollo muy controlado de manera empresarial	Equipo de desarrollo con capacidad de cambio	Equipo conformado de Pocas personas	Puede usarse de manera empresarial, como de manera individual o paralela
Retroalimentación	No hay ningún tipo de retroalimentación, es un modelo rígido	Puede haber retroalimentación es si el usuario lo cree conveniente	Alta volatilidad del entorno, por lo cual una retroalimentación es muy probable	Este modelo se basa en interacciones constantes de retroalimentación
Tiempo de Entrega	Tiempos controlados	Plazos de entregas muy cortos	Periodos de desarrollo cortos	Depende del tiempo de estabilización
Pruebas	Muy controladas con el cliente	Pruebas en forma de Esbozos	Pruebas en base a las funcionalidades	Se hace un testeado luego de concluido el proyecto en su totalidad

(Amaya, 2013)

Este trabajo tiene como objetivo, diseñar una aplicación móvil para la notificación de eventos de una plataforma MOOC, para lo cual es necesario implementar un curso sobre fundamentos de IoT en una plataforma MOOC, para lo cual es importante primero, que son realizar la creación de contenidos del curso “Fundamentos de IOT”, instalar y configurar una plataforma MOOC, implementar el curso “Fundamentos de IOT” utilizando una metodología de MOOC y desarrollar la app de notificación de actividades y fechas importantes del curso “Fundamentos de IOT”.

2. METODOLOGÍA

2. METODOLOGÍA

Para el desarrollo de este proyecto se ha establecido la aplicación de dos metodologías: una para la construcción de la aplicación móvil y otra para el diseño del curso MOOC. A continuación se explica cada una de ellas.

2.1. Metodología Mobile-D

Se ha visto que la metodología ideal para el desarrollo de aplicaciones móviles es la metodología Mobile-D. Se ha elegido esta metodología debido a que se adapta a las herramientas de Android Studio, a la plataforma MOOC, y a las necesidades del producto final, el cual será la aplicación con las notificaciones de eventos del curso MOOC.

Esta metodología consta de cinco fases:

2.1.1. Fase 1: Exploración

El equipo de desarrollo debe generar un plan y establecer las características y los conceptos básicos que están alrededor de todo el proyecto, esta fase se la realiza en 3 etapas las cuales son:

- Establecimiento de interesados. En esta etapa, se identifica y establece cuáles serán los grupos interesados que se involucrarán en los diferentes roles del proyecto. Estos grupos deberán tener amplia variedad de conocimientos y capacidad de cooperación necesaria para una planificación efectiva y de manera controlada con el producto del software. Para ello se establece los actores con el propósito de reconocer que el grupo encargado de la aplicación cuenta la experiencia y conocimiento tanto del dominio como de los requisitos para el desarrollo del producto software.
- Alcance: Se definen los objetivos con respecto a la entrega del producto final, para esto se debe realizar una planeación inicial del proyecto y definir los requisitos iniciales para establecer del alcance real del producto, su propósito y funcionalidad.
- Establecimiento de proyectos: Se designan los recursos para el inicio del proyecto, para esto se definirá la arquitectura que tendrá el mismo

2.1.2. Fase 2: Inicialización del Producto

En esta fase se preparará e identificará todos los recursos necesarios, en esta fase se detalla la planificación y se desarrolla el software para su posterior publicación. Para esto se tendrá que inicializar el proyecto, su propósito es la

preparación de los medios físicos y técnicos necesarios para el proyecto, así como su seguimiento. Para ello se realizará la configuración del entorno definiendo el ambiente de producción, su formación y estableciendo la comunicación con el cliente.

Luego se deberá plantear la planificación inicial cuyo propósito es obtener una buena comprensión de lo que se planea desarrollar, teniendo en cuenta una planificación de la arquitectura, que debe ser integrada según las especificaciones obtenidas y analizando requisitos iniciales, dando prioridad a estos y construyendo nuevos elementos que serán de mayor importancia en el Sistema (Gómez & Hernández, 2016).

2.1.3. Fase 3: Producción

Se repite iterativamente la programación hasta efectuar la implementación de las funcionalidades que el desarrollador necesite. Se inicializa con la construcción de la aplicación, para ser testada al finalizar con la implementación. Esta etapa se divide en 3 fases.

- Fase de la planificación: Se selecciona y planifica el contenido de trabajo y se determina cuáles serán las pruebas de aceptación.
- Fase de trabajo: Se implementa la funcionalidad del sistema previsto. Se inicializa con la programación en parejas, siguiendo con la integración continua del sistema y la información del sistema al cliente.
- Fase de lanzamiento: Elaboración de una versión totalmente funcional del sistema en desarrollo, donde se realiza la integración del sistema, así como la definición de las pruebas a realizarse.

2.1.4. Fase 4: Estabilización

Se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funcione correctamente. Se realiza tareas iterativas muy similares a las realizadas en la fase de producción, con el detalle de que estas labores estarán enfocadas en la integración del sistema con los requisitos del proyecto.

2.1.5. Fase 5: Pruebas

Se prueba y repara el sistema. Se pasa una fase de test hasta tener una versión estable según lo establecido anteriormente por el cliente. El objetivo final de las mismas es obtener un producto que esté disponible, y con funcionalidad íntegra, además que estas pruebas determinarán junto con el cliente si se cumplieron los requisitos del mismo. (Gómez & Hernández, 2016).

2.2. Metodología XMOOC

La metodología de un curso MOOC es dependiente de la tipología y del modelo de aprendizaje a desarrollar. Para el desarrollo del curso MOOC se ha elegido la metodología xMOOC para plataformas abiertas, debido a que se requiere que el curso sea en un entorno abierto, libre, de colaboración continua y de conexión con los involucrados

Esta metodología consta de cuatro fases:

2.2.1. Fase 1: Tipo de curso y plataforma

El primer paso para crear un curso MOOC es definir el tipo de curso y la plataforma dónde se va a desarrollar.

2.2.2. Fase 2: Diseño del curso

Una vez se ha decidido el tipo de curso y la plataforma en la que se va a trabajar, se deberá concretar el diseño del curso, el cual dependerá directamente de todos los recursos que se van a utilizar, sean pedagógicos y tecnológicos. También se definirá la elección definitiva de la plataforma MOOC en base a las necesidades del profesor, la delimitación del tiempo de terminación del Curso MOOC y la metodología pedagógica de cada Docente

2.2.3. Fase 3: Estructura, contenido y secuencias didácticas

A partir del diseño general del curso, es menester empezar a trabajar en los contenidos, su creación y estructuración. Para lo cual es necesario que si se va a implementar un curso en una plataforma MOOC de manera interactiva, se requiere antes los bocetos de lo que serán las actividades, contenido interactivo y multimedia, así como los talleres y las evaluaciones para luego plasmarlas en lo que será nuestro curso MOOC. (Pernías & Luján, 2013).

2.2.4. Fase 4: Evaluación y Canales de comunicación y nivel interacción

El sistema de evaluación de un MOOC también está determinado por el tipo de curso y la herramienta. Así también el docente deberá de considerar su metodología de evaluación antes de elegir con que plataforma va a realizar los criterios de calificación que implementara

En cualquier tipo de curso virtual, es necesario establecer los canales de comunicación, estos canales pueden ser internos o externos, en este caso veremos que en la mayoría del curso MOOC, estos canales serán externos. Debido a la facilidad de que estos se puedan completar fácilmente en línea

con tiempos holgados, se facilitaran las actividades planteadas, tanto la calificación de parte del docente, como el desarrollo del material didáctico de parte del alumnado.

El profesorado que participe en la creación y el diseño de un curso MOOC, deberá contar con algunas competencias digitales y didáctica, en cambio el alumnado se compromete a ser proactivo, completar las tareas, evaluaciones, talleres, así como observar y estudiar atentamente todos los recursos con los que cuenta para su instrucción (Pernías & Luján, 2013).

3. RESULTADOS Y DISCUSIÓN

3. RESULTADOS Y DISCUSIÓN

El sistema consiste en la conexión de una aplicación móvil con la base de datos de un curso MOOC, y el recordatorio de eventos importantes mediante notificaciones. Para el modelo de notificaciones se usa un servicio iterativo con la base de datos para mostrar las notificaciones requeridas cuando el sistema se inicie.

En el desarrollo del sistema se usa el lenguaje de programación Java (Android Studio) ya que es un lenguaje adaptable a las aplicaciones móviles, además de contar con librerías y métodos ideales para nuestra aplicación.

En el desarrollo del sistema se tomó como base la metodología Mobile-D, ya que es una metodología ágil, además de adaptarse a proyectos estudiantiles tal y como lo es nuestra aplicación.

3.1. Aplicación Móvil

3.1.1. Fase 1: Exploración

En la tabla 4 se definen las tareas realizadas en el sprint correspondiente estableciendo las fechas de revisión en cada módulo.

Tabla 4. Fase 1 Exploración

Fase 4. Tiempo: 5 días		Fecha de Cumplimiento: 1/05/2020 – 5/05/2020	
Tarea	Tiempo	Encargado	
Conjunto de Requisitos iniciales	2 días	Antonio Iglesias	
Definición de la Arquitectura	3 días	Antonio Iglesias	

La aplicación de esta fase de desarrollo permitió recopilar información sobre notificaciones que se ejecuten por medio de una tarea Asíncrona, la cual informará las actividades que tengamos de un curso específico, después de ser iniciada la aplicación de notificación. Además se recopilará información del establecimiento de la conexión de la aplicación, con una base de datos MySQL, la cual es contenida por el servicio de la plataforma OpenEDX

Conjunto de Requisitos iniciales

- Se requiere desarrollar una aplicación informativa que notifique al usuario eventos importantes en torno al Curso MOOC “FUNDAMENTOS DE INTERNET DE LAS COSAS (IoT)”
- Estas notificaciones aparecerán en el menú Android deslizable en el extremo superior de la pantalla, y se considerará que contendrá información sobre el evento y la fecha del mismo
- Además se prevé que la aplicación contenga información fundamental del Curso en su pantalla principal
- Como el sistema es informativo el usuario solo deberá iniciarlo por una única vez, así las notificaciones se iniciarán, inclusive si la aplicación fuera cerrada
- El sistema tendrá que ser sencillo, además de tener colores amigables e intuitivos con respecto a la marca institucional del Cliente (Universidad UTE)

Definición de la Arquitectura

Características Notificaciones Y Servicios

Una notificación es un mensaje que Android genera al margen de la interfaz gráfica de una aplicación, que es capaz de recordar a un usuario eventos importantes, correos y mensajes, asimismo información pertinente de la aplicación. El usuario tiene la posibilidad, si el desarrollador lo habilita, de apretar la notificación para realizar una acción directamente desde ella o abrir la app en cuestión.

En cambio un servicio es un componente de una aplicación que puede realizar operaciones de larga ejecución en segundo plano y que no dependerá de la interfaz gráfica. Así otro mecanismo de la aplicación puede inicializar un servicio y continuar operando en segundo plano, aunque la persona que use el terminal móvil realice otros procesos. A la vez, un módulo de la aplicación puede conectarse con un servicio para compartir información y comunicación de procesos.

Junto con estas dos características se desea lograr es que el usuario tenga interacción directa con el sistema. Además que, al iniciar la aplicación, esta se ejecute en primer plano mientras las notificaciones se componen en un segundo plano. En la figura 1 se define el diagrama de casos de uso del módulo de acceso al sistema.

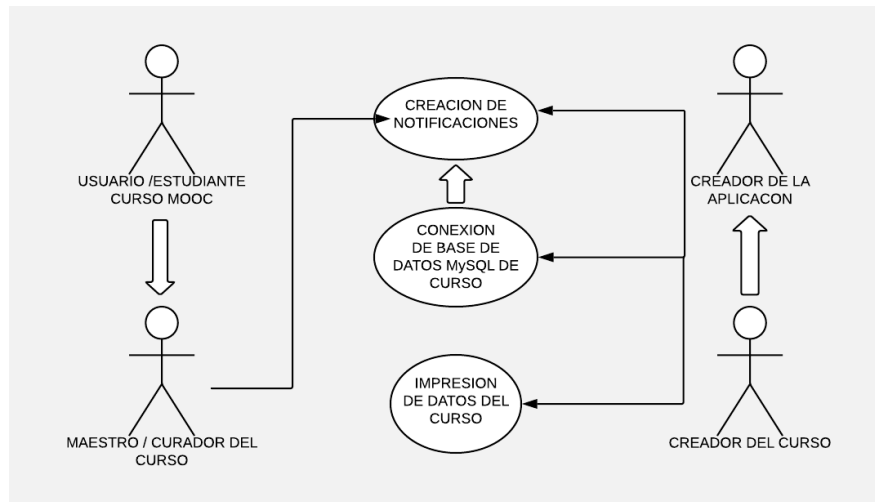


Figura 1. Diagrama de casos de Uso 1.

En la tabla 5 se describen las acciones del único caso de uso a darse en el sistema

Tabla 5. Caso de Uso No 1

Caso de Uso: Inicio de Recordatorio en forma de Notificación	
Actor:	Usuario
Propósito:	El Usuario deberá iniciar una sola vez la aplicación para ver las actividades que tendrá que realizar en su Curso MOOC, además de información fundamental del mismo
Eventos del Sistema	
Acción	Respuesta del Sistema
1. Iniciar la Aplicación	<ul style="list-style-type: none"> Se iniciara el servicio, el cual hará que una tarea Asíncrona genere en un tiempo específico una Notificación con los eventos del curso como tal al conectarse con MySQL (la base de datos del curso).

3.1.2. Fase 2: Inicialización Del Producto

En la tabla 6 se definen las tareas realizadas en el sprint correspondiente estableciendo las fechas de revisión en cada módulo.

Tabla 6. Fase 2 Inicialización del Producto

Fase 2. Tiempo: 6 días	Fecha de Cumplimiento: 6/05/2020 – 11/05/2020	
Tarea	Tiempo	Encargado
Planeamiento de la Arquitectura	1 día	Antonio Iglesias
Análisis de Requerimientos Funcionales	5 días	Antonio Iglesias

Puesta en Marcha del Proyecto

Se plantea al inicio de esta fase, que el entorno productivo en el que se desarrollará el proyecto será Android Studio, realizando las pruebas en una máquina virtual que simula un entorno Móvil con el Sistema Operativo Android. La comunicación con el cliente se ha establecido de manera que, por falta de disponibilidad por parte del cliente, el mismo ha dispuesto que, una vez completado la totalidad del desarrollo del proyecto y su integración con los medios requeridos, se revisará en su totalidad la funcionalidad del mismo.

Planeamiento de la Arquitectura

Para la clase My Service, se planea implementar los siguientes métodos:

URLConnection

Inicialmente se desea obtener un nuevo `URLConnection` llamando `URLConnection.openConnection()` y se envía lo obtenido a `URLConnection`.

Luego se prepara una solicitud, por medio de una de sus propiedades básicas y elementales la cuales son las URI. Los encabezados de las solicitudes también se pueden usar para adquirir información como los datos de las base de datos de My SQL de nuestro curso.

Después el sistema se encarga de leer la respuesta, la cual incluye los metadatos necesarios para que la sesión de la conexión se establezca, se utiliza el método `URLConnection.getInputStream()`. El cual analiza el cuerpo de la conexión si es necesario.

Finalmente luego de haberse leído el cuerpo, se procederá a desconectar una vez se ha leído las respuesta, esta libera los recursos de la conexión, la cual espera reusarse.

Async Task

Los métodos de la tarea Asíncrona a usarse son:

`DoInBackground (Params...)`: Es el subproceso el cual realiza los cálculos en segundo plano que ingresan en la tarea asíncrona, los parámetros de dicha tarea se van a este paso, el cálculo devuelto será llevado al último paso, en este proceso se podrá un hilo el cual dará los tiempos que se requiere funcione la aplicación.

`OnPostExecute (Result)`: Este método el cual es llamado en un subproceso que va a funcionar en segundo plano independientemente de la interfaz gráfica de la aplicación, luego que finaliza el cálculo hecho en segundo plano como el método anterior, el resultado del mismo es analizado como un parámetro.

Service

Para la creación de un servicio, se usará una clase ya existente, anulando métodos de devolución de llamada, el cual maneja aspectos importantes del ciclo de vida del servicio:

`OnStartCommand ()`

Este método invoca a `StartService ()`, el cual solicita que se inicie el servicio, una vez que esté en ejecución, este entra en segundo plano y se ejecuta de manera indefinida, ideal al iniciar la aplicación por medio de un recordatorio.

`OnCreate ()`

En este procedimiento se realizan procedimientos únicos con respecto a la configuración donde el servicio es creado por primera vez, si el servicio se está ejecutando este método no es llamado.

`OnDestroy ()`

Este método es ideal para cuando se requiere que el servicio en cuestión se pare y se destruya, en este caso se puede usar para liberar recursos de las notificaciones usados por nuestro Async Task.

Calendar

La clase `calendar` es una clase de tipo abstracta, la cual convierte por medio de métodos, el tiempo específico que puede ser capturado en tiempo real, a Años, Meses, Días e inclusive horas si así se lo requiere, con esto es posible manipular varios campos, obtener los mismos y que estos ayuden a las elecciones que realizara nuestro programa. Gracias al método `getInstance ()`, que capturará la fecha en tiempo real, la misma que será almacenada en un `SimpleDateFormat` comparara las consultas realizadas con datos específicos del sistema.

Análisis de Requerimientos Funcionales

- Al observar que se requiere notificaciones que informen de actividades que se almacenan en una base de datos, se ha determinado que se realizará una conexión con una base de datos, localmente para realizar pruebas, y luego se conectara a un Web Service que contenga la base de datos en sí misma.
- Al requerir que las notificaciones aparezcan sin interrumpir actividades que el usuario este realizando en su dispositivo móvil, se ha decidido que las mismas sean una tarea en segundo plano de la aplicación y que las mismas aparezcan en un periodo determinado de tiempo.
- Al prever que se requiere extraer información para proyectarla en nuestras Aplicación, se utilizará la conexión obtenida anteriormente de la base de datos contenida en un Web Service para la obtención de datos y procesamiento de la información del curso.
- Al iniciar el sistema, el usuario no debe realizar ninguna actividad adicional para generar las notificaciones como para obtener la información requerida.
- El sistema tendrá que ser sencillo, además de tener colores amigables e intuitivos con respecto a la marca institucional del Cliente (Universidad UTE)

3.1.3. Fase 3: Producción

En la tabla se definen las tareas realizadas en el sprint correspondiente estableciendo las fechas de revisión en cada módulo.

Tabla 7. Fase 3 Producción

Fase 3. Tiempo: 10 días		Fecha de Cumplimiento: 12/05/2020 – 22/05/2020	
Tarea	Tiempo	Encargado	
Generación de pruebas de Aceptación	5 días	Antonio Iglesias	
Inicio de Interacciones e implementación de métodos requeridos	5 días	Antonio Iglesias	

Generación de pruebas de Aceptación

En la aplicación de esta fase se ha decidido que el sistema tendrá 4 criterios en sus pruebas de Aceptación:

- Conexión de Aplicación con Base de datos Local en donde se realiza la conexión y verificación de funcionalidad de DownloadUrl con el fin de obtener datos de Base de datos Local MySQL de OpenEDX
- Generación de Notificaciones con datos de Base de datos local en el cual se genera una notificación recordando un evento en la base de datos local para generar notificaciones como una actividad en segundo plano, además de ejecutarlas pese a que la aplicación sea destruida
- Conexión de Aplicación con Base de datos en Web Service, aquí se realiza la conexión y verificación de funcionalidad de DownloadUrl y se obtiene datos de Base de datos en Web Service MySQL de OpenEDX
- Generación de Notificaciones con datos de Base de datos Web Service donde se genera una notificación recordando un evento en la base de datos Web Service MySQL de OpenEDX con el objetivo de generar notificaciones como una actividad en segundo plano, además de ejecutarlas pese a que la aplicación sea destruida

Inicio de Interacciones e implementación de métodos requeridos

Se planifica 10 días de interacciones divididas en Sprint de 5 días cada uno, en los cuales se desarrollara la implementación de los métodos descritos en la planificación de la arquitectura, en donde los responsables realizaran la integración de todos estos métodos en la tarea Asíncrona detallada anteriormente.

Diseño de la interfaz

Para la elaboración de la interfaz de se han tomado en cuenta las siguientes características:

- Simplicidad: interfaz que sea completamente ligera, debido a que es una app de corte informativo, el usuario solo tendrá que iniciar la aplicación para acceder al servicio requerido
- Practicidad: interfaz que notifica al usuario de sus actividades mediante notificaciones que no interrumpan el uso del sistema móvil.
- Estímulo: interfaz llamativa, la cual permite al usuario adquirir interés por el producto que se está conectando con la aplicación.

Para el diseño se tomó en cuenta interfaces informativas con conexiones simples a bases de datos, las cuales permite al usuario observar información por la cual instala la aplicación en su dispositivo. En la figura 2 se muestra un prototipo de la interfaz principal:



Figura 2. Prototipo Interfaz de Aplicación

Desarrollo

Para el desarrollo de la aplicación se iniciaran Sprint de 10 días, los cuales se dividirán en 2 fases:

- Inicio de Interacciones e implementación de métodos requeridos
- Desarrollo de la parte global de la aplicación

Método de Conexión a la Base MySQL

Para la conexión a My SQL (base de datos), fue necesario implementar el método DownloadUrl, el cual gracias a un método Get capta la dirección URL en la que está contenida una consulta en lenguaje PHP, y devuelve el resultado del procesamiento de la misma. En la figura 3 se describe el algoritmo para llamar al método DownloadUrl.

```
private String downloadUrl(String myurl) throws IOException {
    Log.i( tag: "URL", msg: ""+myurl);
    myurl = myurl.replace( target: " ", replacement: "%20");
    InputStream is = null;
    // Only display the first 500 characters of the retrieved
    // web page content.
    int len = 500;

    try {
        URL url = new URL(myurl);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setReadTimeout(10000 /* milliseconds */);
        conn.setConnectTimeout(15000 /* milliseconds */);
        conn.setRequestMethod("GET");
        conn.setDoInput(true);
        // Starts the query
        conn.connect();
        int response = conn.getResponseCode();
        Log.d( tag: "respuesta", msg: "The response is: " + response);
        is = conn.getInputStream();

        // Convert the InputStream into a string
        String contentAsString = readIt(is, len);
        return contentAsString;

        // Makes sure that the InputStream is closed after the app is
        // finished using it.
    } finally {
        if (is != null) {
            is.close();
        }
    }
}
```

Figura 3. Algoritmo método DownloadUrl

Luego se deberá implementar este método en una Tarea Asíncrona, para que se ejecute inicialmente en el momento de inicio de la tarea, luego se formará la tarea posterior, que almacena los datos que devuelve la consulta que capta el método DownloadUrl por medio de una URL ya descrita anteriormente, a esta tarea la llamaremos ConsultarDatos. En la figura 4 se describe el algoritmo ConsultarDatos donde se llama al método DownloadUrl y se almacena los datos consultados y almacenados en un arreglo JSON.

```
private class ConsultarDatos extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... urls) {
        // params comes from the execute() call: params[0] is the url.
        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may be invalid.";
        }
    }
    // onPostExecute displays the results of the AsyncTask.
    @Override
    protected void onPostExecute(String result) {
        JSONArray ja = null;
        try {
            ja = new JSONArray(result);
            etId.setText(ja.getString( index: 0));
            etCreated.setText(ja.getString( index: 1));
            etModified.setText(ja.getString( index: 2));
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}
```

Figura 4. Algoritmo ConsultarDatos

Luego se implementará un Service el cual se dividirá en 3 partes fundamentales

OnCreate: El cual llamara a crear al servicio como tal para su ejecución en segundo plano. Además de llamar a una clase Timer que controlara que esta tarea, una vez iniciada la aplicación por primera vez, se ejecute en segundo plano a pesar de que esta se destruya. En la figura 5 se describe el algoritmo que contiene el método OnCreate descrito anteriormente.

```
@Override
public void onCreate() {
    super.onCreate();
    mTimer = new Timer();
    mTimer.schedule(timerTask, delay: 2000, period: 2 * 1000);
}
```

Figura 5. Algoritmo Método OnCreate.

OnStartCommand: En este método llama a otros métodos almacenados en nuestro servicio, en este caso se llamará a nuestra tarea Asíncrona con un recurso URL, que permite la conexión de la base de datos como se explicó anteriormente, además del método START_STICKY. En la figura 6 se describe el algoritmo que contiene el método OnStartCommand descrito anteriormente

```

@Override
public int onStartCommand(Intent intent, int flags, int startId) {

    // TODO Auto-generated method stub
    //new MyService.ConsultarDatos().execute("http://10.0.3.2/CursoAndroid/consulta.php?id=2");
    new MyService.ConsultarDatos().execute("http://archivosphpopenedx.000webhostapp.com/consulta.php?display_name=FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)");
    return android.app.Service.START_STICKY;
}

```

Figura 6. Algoritmo Método OnStartCommand.

OnDestroy: En este método se destruyen los métodos como mTimer y timerTask, pero gracias a un Intent que ejecuta de nuevo nuestra aplicación junto con un Broadcast, la aplicación estará lista para ejecutar estos métodos de manera ininterrumpida una vez se ejecute por primera vez. En la figura 10 se describe el algoritmo que contiene el método OnDestroy descrito anteriormente

```

@Override
public void onDestroy() {
    try {
        mTimer.cancel();
        timerTask.cancel();
    } catch (Exception e) {
        e.printStackTrace();
    }

    Intent intent = new Intent( action: "com.android.openedxnotificacion");
    intent.putExtra( name: "yourvalue", value: "torestone");
    sendBroadcast(intent);
}

package com.example.openedxnotificacion;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.util.Log;

public class ReceiverCall extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        Log.i( tag: "Service Stops", msg: "Ohhhhhhh");
        context.startService(new Intent(context, MyService.class));
    }
}

```

Figura 7. Algoritmo Método OnDestroy.

Módulo MyService.java

En la figura 8 se describe el diagrama de flujo de procesamiento de notificaciones del sistema, mediante el ingreso de recurso URL para impresión de consultas y notificaciones de eventos. Ver Anexo 2, que contiene el algoritmo usado para su funcionalidad.

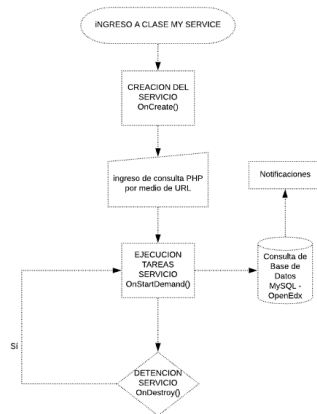


Figura 8. Diagrama de flujo de procesamiento de notificaciones.

Módulo MainActivity

En la figura 9 se describe el diagrama de flujo de ingreso de información al sistema, mediante la ejecución de una consulta que refleje los datos de la base MySQL, además de la llamada que ejecuta el Service. Ver Anexo 1, que contiene el algoritmo usado para su funcionalidad.

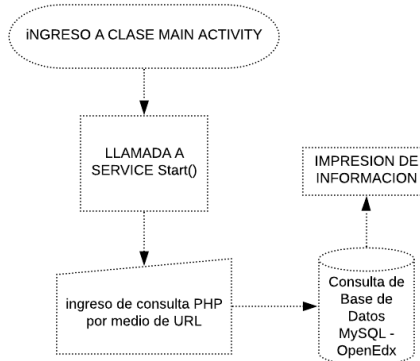


Figura 9. Diagrama de flujo de ingreso al sistema.

Módulo ReceiverCall

En la figura 10 se describe el diagrama de flujo de ingreso al método Receiver Call por medio de su misma clase mediante la llamada de MyService, además de la llamada que ejecuta el Service. Ver Anexo 3, que contiene el algoritmo usado para su funcionalidad.

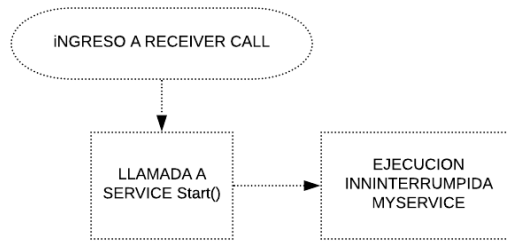


Figura 10. Diagrama de flujo de ingreso a ingreso al método Receiver Call.

3.1.4. Fase 4: Estabilización

En la tabla 8 se definen las tareas realizadas en el sprint correspondiente estableciendo las fechas de revisión en cada módulo realizado.

Tabla 8. Fase 4 Estabilización

Fase 4. Tiempo: 10 días	Fecha de Cumplimiento: 23/05/2020 – 2/06/2020	
Tarea	Tiempo	Encargado
Afinamiento de la aplicación con Base de Datos MySQL en Web Service	5 días	Antonio Iglesias
Afinamiento notificaciones Push	5 días	Antonio Iglesias

Para la aplicación de la fase de estabilización del programa se iniciará un único Sprint de 10 días, los cuales se dividirán en 2 fases:

- Afinamiento de la aplicación con Base de Datos MySQL en Web Service
- Afinamiento notificaciones Push

Afinamiento de la aplicación con Base de Datos MySQL en Web Service

Debido a que la base de datos que se desea conectar con la aplicación, está contenida en un web Service, se ha decidido que el archivo en formato PHP que realizara la consulta, este alojado en un mismo Web Service para ser llamado mediante un objeto URL ya definido anteriormente

En las Figuras 11, 12 y 13 se pueden ver los códigos en PHP utilizado para acceder al mismo, además del código de la consulta con el objeto URL

```

<?php
header( 'Content-Type: text/html;charset=utf-8' );

function ejecutarSQLCommand($commando){

    $mysqli = new mysqli("db4free.net", "antonioiglesias", "liomessil017", "tutorial27");

    /* check connection */
    if ($mysqli->connect_errno) {
        printf("Connect failed: %s\n", $mysqli->connect_error);
        exit();
    }

    if ( $mysqli->multi_query($commando) ) {
        if ($resultset = $mysqli->store_result() ) {
            while ($row = $resultset->fetch_array(MYSQLI_BOTH) ) {
                }
            $resultset->free();
        }
    }

    $mysqli->close();
}

function getSQLResultSet($commando){

    $mysqli = new mysqli("db4free.net", "antonioiglesias", "liomessil017", "tutorial27");
    /* check connection */
    if ($mysqli->connect_errno) {
        printf("Connect failed: %s\n", $mysqli->connect_error);
        exit();
    }

    if ( $mysqli->multi_query($commando) ) {
        return $mysqli->store_result();
    }

}

$mysqli->close();

```

Figura 11. Código conexión de Aplicación a Base de datos MySQL.

```

<?php
include('functions.php');
$id=$_GET["id"];

if($resultset=getSQLResultSet("SELECT * FROM `course_overviews_courseoverview` WHERE id='$id'")){
    while ($row = $resultset->fetch_array(MYSQLI_NUM)){
        echo json_encode($row);
    }
}
?>

```

Figura 12. Código consulta de Aplicación a tablas de Base de datos MySQL.

```

new ConsultarDatos().execute("http://archivosphpopenexa.600webhostapp.com/consulta.php?display_name=FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)");

```

Figura 13. Consulta de aplicación a Base de datos por medio de código PHP.

Perfeccionamiento notificaciones tipo Push

Se ha decidido integrar un objeto calendar, el cual captura en el formato de fecha de la base de datos, es decir “yyyy-MM-dd”, luego se compara con la fecha extraída por la consulta en sí, y si esta comparación es correcta, entonces se mostrará una notificación dependiendo el tiempo que el desarrollador considere prudente, así realizándolo para las notificaciones que el sistema requiera. En la figura 14 se describe el algoritmo que contiene el método OnDestroy descrito anteriormente


```

SimpleDateFormat df = new SimpleDateFormat( pattern: "yyyy-MM-dd");
String formattedDate = df.format(C);
//etCreated.setText(ja.getString(1));
//etModified.setText(ja.getString(2));

String fechacreated = ja.getString( index: 1).substring(0,10);
String fechamodified = ja.getString( index: 2).substring(0,10);

if(fechacreated.equals(formattedDate)){
    notificacion.setSmallIcon(R.mipmap.ic_launcher);
    notificacion.setTicker("Nueva Notificacion");
    notificacion.setWhen(System.currentTimeMillis());
    notificacion.setContentTitle("HOY ES EL DIA PARA INSCRIBIRTE EN EL CURSO IoT");
    notificacion.setContentText(fechacreated);

    Intent intent = new Intent ( packageContext: MyService.this,MyService.class);
    PendingIntent pendingintent = PendingIntent.getActivity( context: MyService.this, requestCode: 0,intent,PendingIntent.FLAG_UPDATE_CURRENT);
    notificacion.setContentIntent(pendingintent);
    NotificationManager nm = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
    nm.notify(idunica,notificacion.build());
}

```

Figura 14. Algoritmo Creación de Notificaciones.

3.1.5. Fase 5: Pruebas

En la tabla 9 se definen las pruebas realizadas al sistema de control de inventarios en esta fase, estas pruebas fueron realizadas en un sprint nuevo de 10 días para completarlas.

Tabla 9. Pruebas Funcionales del Sistema

Id	1	2	3	4
Caso de Prueba	Conexión de Aplicación con Base de datos Local	Generación de Notificaciones con datos de Base de datos local	Conexión de Aplicación con Base de datos en Web Service	Generación de Notificaciones con datos de Base de datos Web Service
Descripción	Se realiza la conexión y verificación de funcionalidad de DownloadUrl	Se genera una notificación recordando un evento en la base de datos local	Se realiza la conexión y verificación de funcionalidad de DownloadUrl	Se genera una notificación recordando un evento en la base de datos Web Service MySQL de OpenEDX
Fecha	01/06/2020	01/06/2020	30/06/2020	30/06/2020
Funcionalidad	Obtener datos de Base de datos Local MySQL de OpenEDX	Generar notificaciones como una actividad en segundo plano, además de ejecutarlas pese a que la aplicación sea destruida	Obtener datos de Base de datos en Web Service MySQL de OpenEDX	Generar notificaciones como una actividad en segundo plano, además de ejecutarlas pese a que la aplicación sea destruida
Datos/Acciones de entrada	Iniciar Aplicación	Iniciar Aplicación	Iniciar Aplicación	Iniciar Aplicación
Resultado Esperado	Obtención de datos en pantalla principal de la Aplicación	Obtención de notificación generando evento consultado en la base de datos	Obtención de datos en pantalla principal de la Aplicación	Obtención de notificación generando evento consultado en la base de datos
Requerimientos de Ambiente de Pruebas	Conexión Establecida con Servidor Local mediante la ejecución de consulta PHP	Conexión Establecida con Servidor Local mediante la ejecución de consulta PHP	Conexión Establecida con Servidor Web MySQL de OpenEDX mediante la ejecución de consulta PHP en servidor Web	Conexión Establecida con Servidor Web MySQL de OpenEDX mediante la ejecución de consulta PHP
Dependencias con otros casos de Prueba	Ninguna	Prueba 1	Ninguna	Prueba 3
Resultado Obtenido	Impresión de datos Correctos en los campos establecidos de la pantalla principal	Impresión cada cierto tiempo de notificación de evento consultado en la base de datos	Impresión de datos Correctos en los campos establecidos de la pantalla principal	Impresión cada cierto tiempo de notificación de evento consultado en la base de datos
Estado	Revisado	Revisado	Revisado	Revisado
Ultima Fecha de Estado	15/06/2020	15/06/2020	27/07/2020	27/07/2020

3.2. Metodología XMOOC

En el desarrollo del curso especificado en la plataforma MOOC se ha elegido la metodología XMOOC, la cual es una sistemática que genera conexión con el estudiante y el docente desarrollador del curso.

3.2.1. Fase 1: Tipo de curso y plataforma

En la tabla 10 se definen las tareas realizadas en el sprint correspondiente estableciendo las fechas de revisión en cada módulo.

Tabla 10. Fase 1: Tipo de curso y plataforma

Fase 1. Tiempo: 12 días	Fecha de Cumplimiento: 01/03/2020 – 13/03/2020	
Tarea	Tiempo	Encargado
Elección Plataforma MOOCs	5 días	Antonio Iglesias
Lectura documentación global	2 días	Antonio Iglesias
Instalación Plataforma MOOCs	5 días	Antonio Iglesias

En la aplicación de esta fase se ha elegido que sea un curso sencillo, con 5 clases, las cuales serán evaluadas inmediatamente al finalizar la misma, con opción a retroalimentación, ubicando los contenidos en videos y presentaciones interactivas, que cuentan con material Introductorio al mundo de IoT.

En la tabla 11 y figura 15, se observa un cuadro y gráfico comparativo entre las distintas plataformas MOOC, con características planteadas por el desarrollador del curso en beneficio de lo que se requiere en el trabajo de titulación, que determinaran que plataforma será la elegida para el desarrollo del curso MOOC.

Tabla 11. Cuadro comparativo Plataformas MOOC

Plataformas	Miriada	Coursera	Edx
Disponibilidad de Temas	X	X	X
Certificados	X	X	X
Cursos de modalidad Abierta(Gratis)	X	X	X
Estructura amigable	X	X	X
Documentación de libre acceso			X
Base de datos Accesible			X
Entornos de desarrollo libre			X

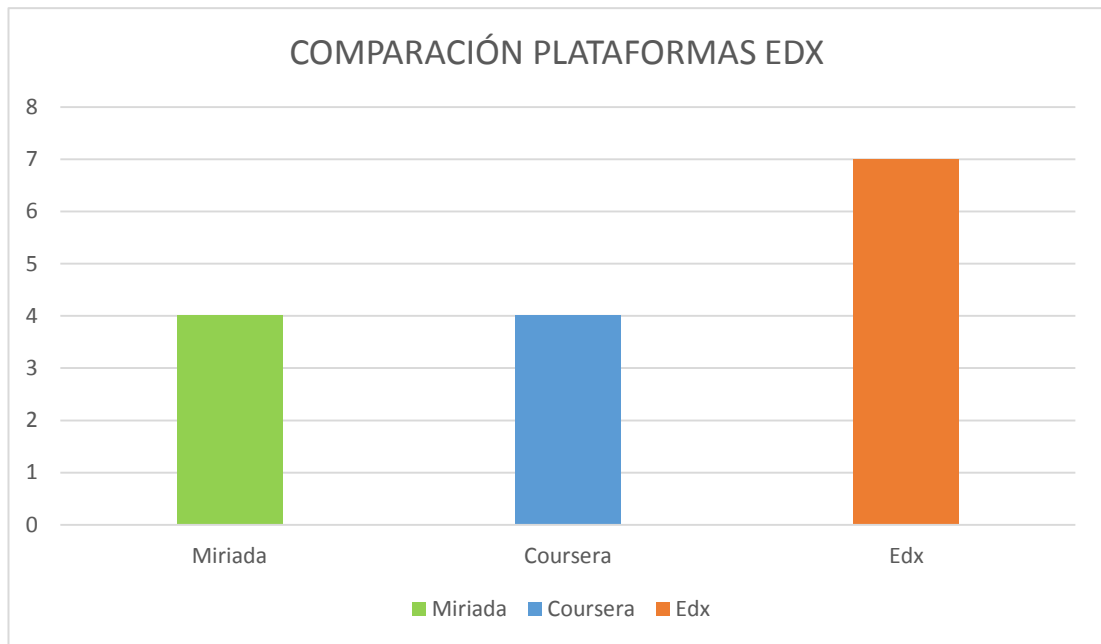


Figura 15. Gráfico comparativo Plataformas MOOC

Después de un análisis de varias plataformas Open Source, se ha decidido que para este curso se ha elegido la plataforma OpenEDX Powered By Bitnami ©, debido a que cuenta con un sistema de acceso libre al desarrollo de cursos de todo ámbito, además de la posibilidad de implementar recursos didácticos y evaluaciones, inclusive cuenta con un gestor que administre esta base de datos MySQL, la cual puede ser migrada a un servidor Web, lo cual nos facilita el control de actividades claves del curso en sí.

Instalación Plataforma OpenEDX en entorno de Desarrollo Debian

- Se descargará el archivo Bitnami-edx-ironwood.2-4-linux-x64-installer.run desde el repositorio oficial de Bitnami. Luego se ejecutará el comando de instalación que se detalla en la figura 16. En la figura 17 se detalla la ventana de ejecución de la plataforma

```
usuario@IGLESIASUTE:~$ ./bitnami-edx-ironwood.2-4-linux-x64-installer.run
```

Figura 16. Comando Ejecución Ironwood.



Figura 17. Pantalla de Instalación Ironwood.

- Luego se elegirá el sitio de la instalación, así como las credenciales con las que podremos administrar esta plataforma y al final la dirección IP que corresponde al HostName. En la figura 18 se detalla el directorio de instalación de la plataforma. Mientras que en la figura 19 se detalla las credenciales elegidas para la autenticación en la misma. Así como en la figura 20 se observa el HostName a utilizar.

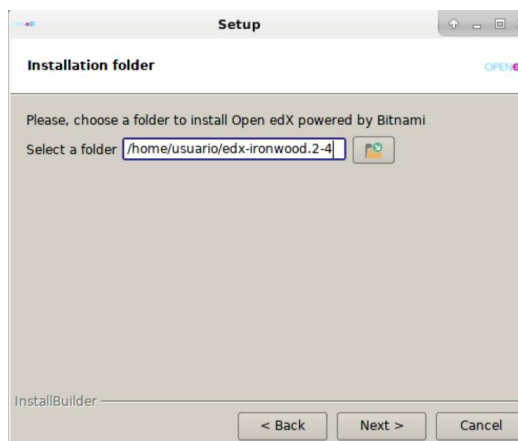


Figura 18. Pantalla de Instalación Ironwood.

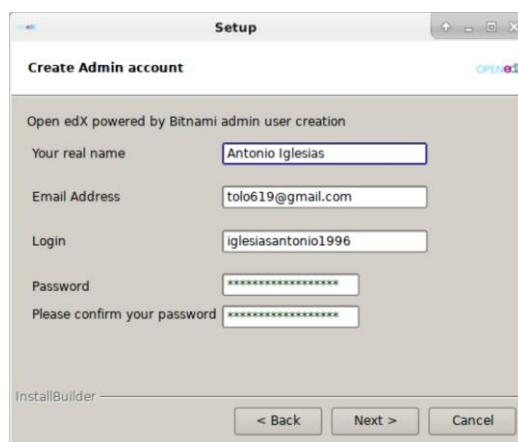


Figura 19. Credenciales Autenticación Ironwood.

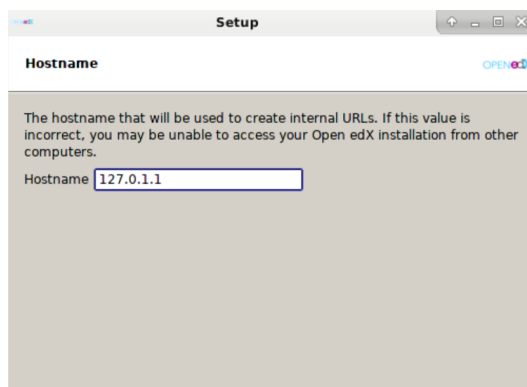


Figura 20. Hostname Ironwood.

- Finalmente se instalará la plataforma y se la iniciará en un entorno local. En la figura 21 se observa la plataforma instalada y en ejecución. En la figura 22 se observa la lista de los Servidores locales con los que cuenta nuestra plataforma, además de varias acciones que podemos realizar con los mismos.



Figura 21. Plataforma OpenEDX Ironwood.

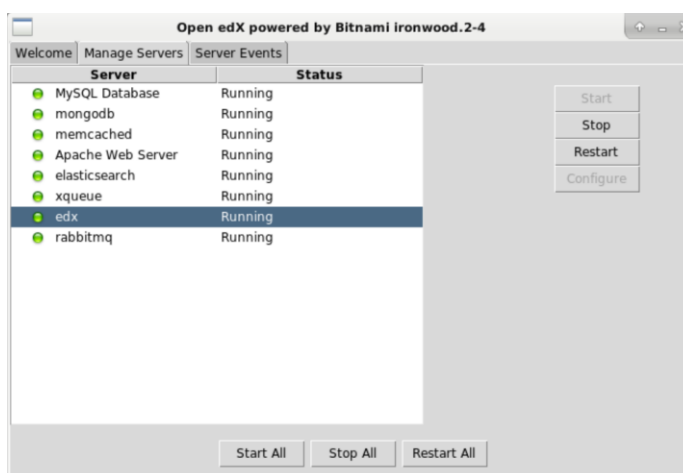


Figura 22. Servidores Plataforma OpenEDX Ironwood.

3.2.2. Fase 2: Diseño Del Curso

En la tabla 12 se definen las tareas realizadas en el sprint correspondiente estableciendo las fechas de revisión en cada módulo realizado.

Tabla 12. Fase 2: Diseño del curso

Fase 2. Tiempo: 12 días	Fecha de Cumplimiento: 13/03/2020 – 27/03/2020	
Tarea	Tiempo	Encargado
Elección recursos a Utilizar	3 días	Antonio Iglesias
Tiempo de Terminación de Curso MOOC	4 días	Antonio Iglesias
Elección Metodología Pedagógica	5 días	Antonio Iglesias

En la aplicación de esta fase se observa que luego de haber elegido el tipo correcto de plataforma MOOC la cual ha sido OpenEDX, se concretará el diseño que es determinado por las siguientes características:

Recursos a utilizar

Este curso por su tipo de metodología, su facilidad de acceso e información, y que será dirigido en gran parte a población mayormente joven, se ha determinado que se utilizarán recursos didácticos que serán videos y presentaciones, los cuales contarán con una descripción detallada del tema, además de diagramaciones y animaciones llamativas para que los usuarios más jóvenes sientan pertenencia con este tipo de contenido multimedia.

Tiempo de terminación del Curso MOOC

Se ha determinado que al ser un curso de Fundamentos meramente introductorios, la duración del curso no será mayor a 1 semana, en la cual se podrá realizar una actividad al día, durante 5 días seguidos, y si no pudiese completar estas actividades, tendrá 2 días para su complementación, o una retroalimentación.

Metodología pedagógica de cada Docente

Según varias metodologías ágiles implementadas inclusive por docentes de esta universidad, se implementará un sistema de evaluación continuo, donde se valorará algunos aspectos fundamentales con respecto a los contenidos expuestos en el material didáctico al final de la sesión de estudio, las evaluaciones darán al estudiante una pauta de si debe de realizar una retroalimentación o proseguir con los siguientes temas expuestos en el curso

3.2.3. Fase 3: Estructura, contenido y secuencias didácticas

En la tabla 13 se definen las tareas realizadas en el sprint correspondiente estableciendo las fechas de revisión en cada módulo realizado.

Tabla 13. Fase 3 Estructura, contenido y secuencias didácticas

Fase 3. Tiempo: 20 días	Fecha de Cumplimiento: 27/03/2020 – 15/04/2020	
Tarea	Tiempo	Encargado
Definición estructura Curso	2 días	Antonio Iglesias
Preparación de Contenidos y secuencias Didácticas (Videos)	16 días	Antonio Iglesias
Subida de Videos a la Plataforma según la estructura del curso	2 días	Antonio Iglesias

Estructura

En la aplicación de esta fase se define que el curso “FUNDAMENTOS DEL INTERNET DE LAS COSAS” contendrá la siguiente estructura:

- INTRODUCCION DE EL INTERNET DE LAS COSAS
 - Video Explicativo
 - Presentación Power Point
 - Presentación Prezi
 - Evaluación
- CONCEPTO DE EL INTERNET DE LAS COSAS
 - Video Explicativo
 - Presentación Power Point
 - Presentación Prezi
 - Evaluación
- CARACTERISTICAS DE EL INTERNET DE LAS COSAS
 - Video Explicativo
 - Presentación Power Point
 - Presentación Prezi
 - Evaluación
- RECOLECCION DE LA INFORMACION DE EL INTERNET DE LAS COSAS
 - Video Explicativo
 - Presentación Power Point
 - Presentación Prezi
 - Evaluación
- APLICACIONES DE EL INTERNET DE LAS COSAS
 - Video Explicativo

- Presentación Power Point
- Presentación Prezi
- Evaluación

En el anexo 9 y 10, se indica las características de la plataforma Studio y LMS en donde se desarrolla este curso.

En la Figura 23 se observa la estructura del curso ya integrada en la plataforma OpenEDX

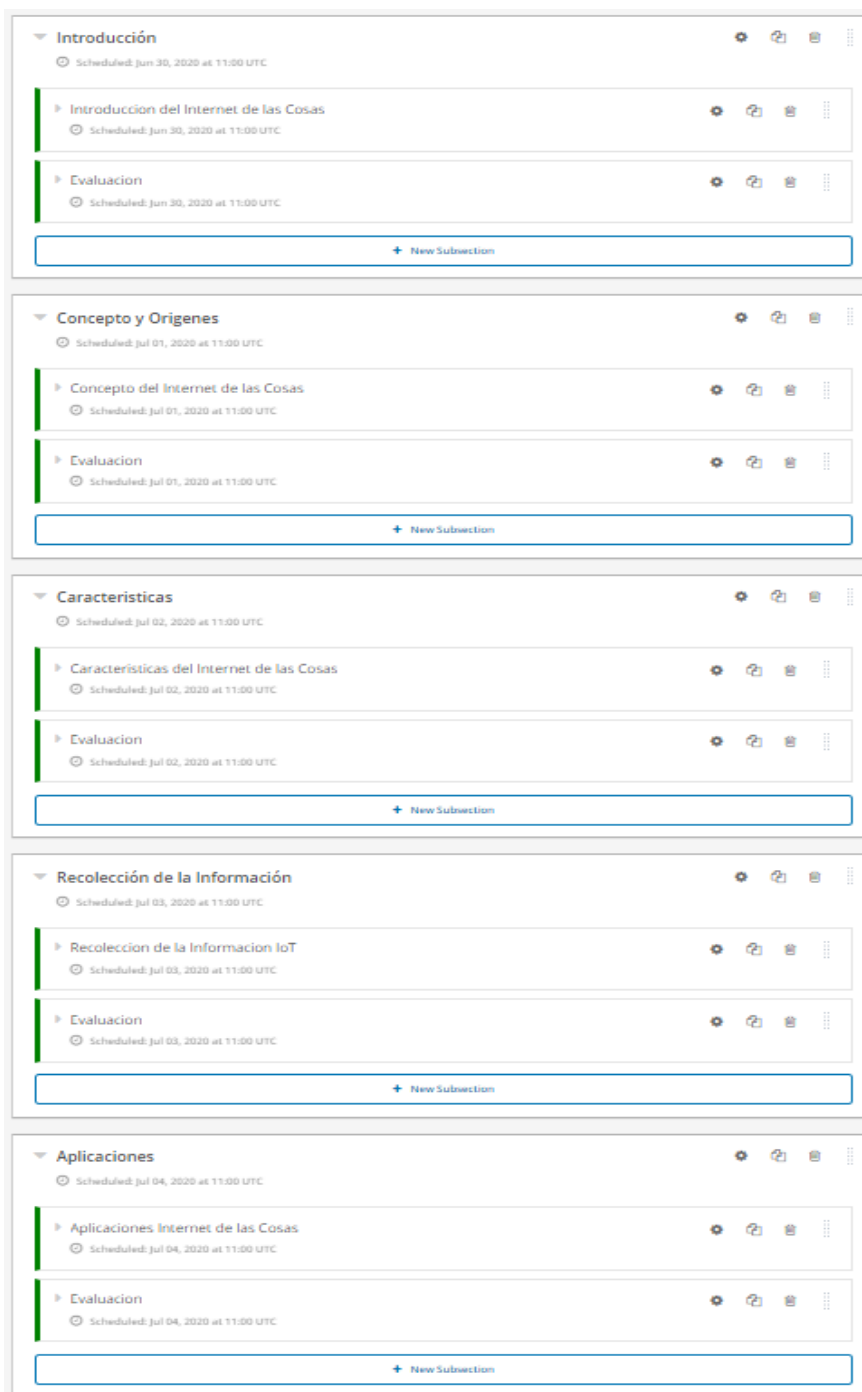


Figura 23. Estructura Curso “Fundamentos Del Internet De Las Cosas (IoT)”.

Contenido y Secuencias Didácticas

El contenido estará reeditado de varias fuentes, las cuales dan orientación al estudiante de las bases introductorias acerca del Internet de las Cosas, además de dar un enfoque general de varias opiniones acerca de esta nueva tecnología, además de brindar un contenido visual multimedia por medio de gráficas y diagramas que ayudan a comprender correctamente lo indicado por medio de la explicación incluida en el video, tal y como se puede visualizar en la figura 24.

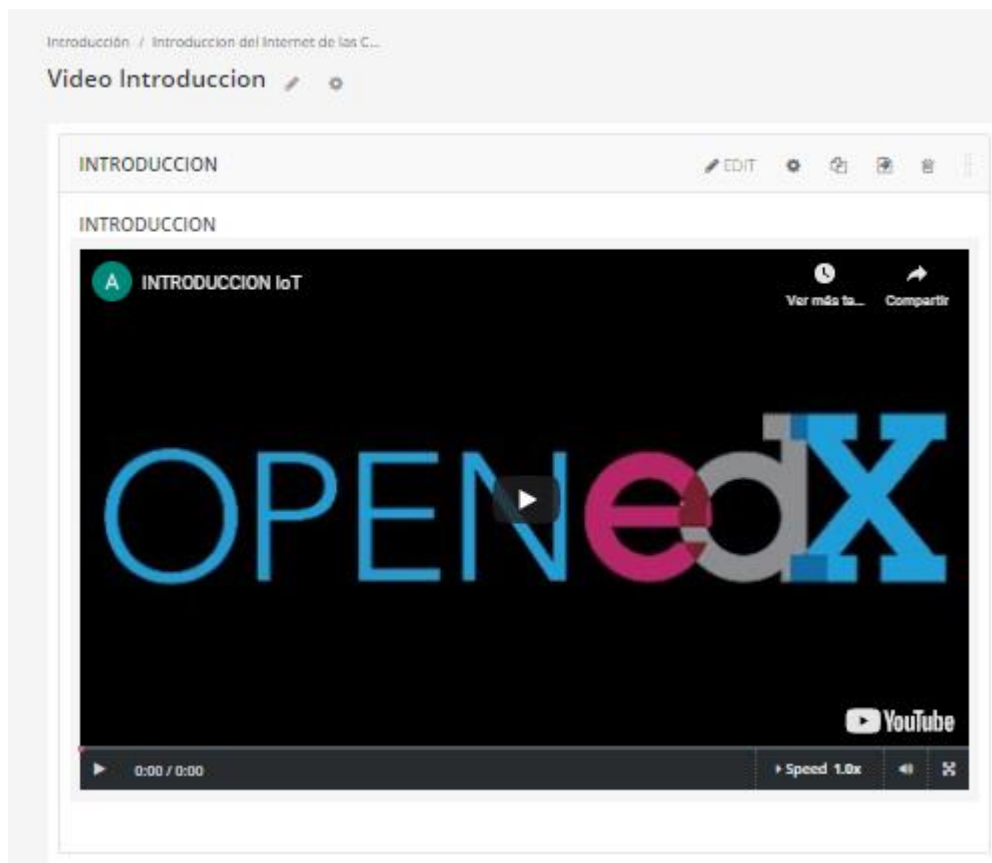


Figura 24. Recurso Didáctico en plataforma OpenEDX.

Videos

Los videos tendrán que estar contenidos en la plataforma de YouTube. Como se observa en la figura 25, estos podrán tener transcripción de texto siempre y cuando el docente agregue la misma.

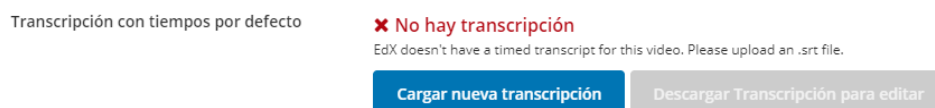


Figura 25. Transcripción Videos en plataforma OpenEDX.

Además se recomienda que para la correspondiente subida a YouTube, el video debe estar en formato .mp4. Luego se crea una nueva unidad como se

observa en la figura 26 y se añade el componente de Video como se observa en la figura 27. Se edita la información del video, teniendo en cuenta que se debe ingresar el link del mismo en la casilla “URL de video por defecto” como se observa en la figura 28.

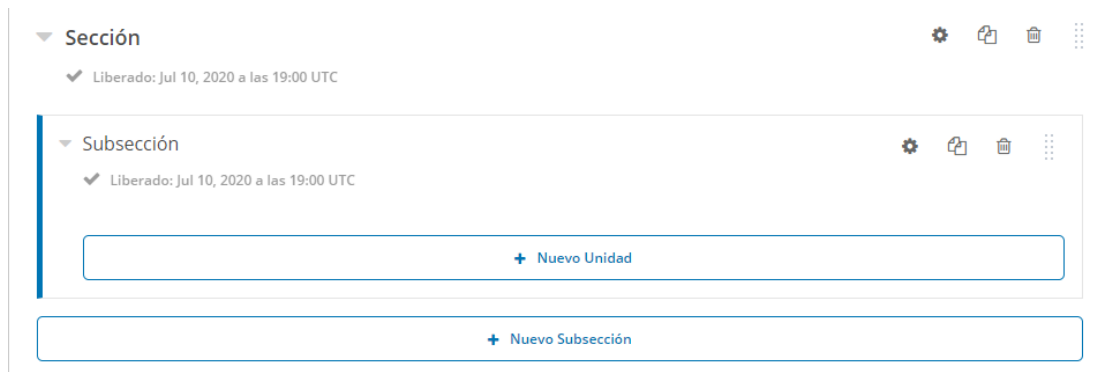


Figura 26. Nueva Unidad de Curso en plataforma OpenEDX.

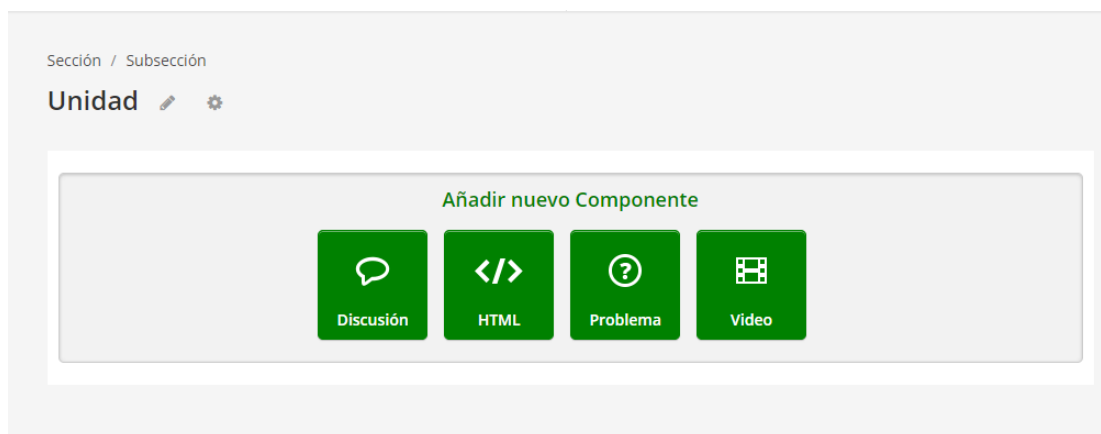


Figura 27. Componente de Video en Unidad de Curso en plataforma OpenEDX.

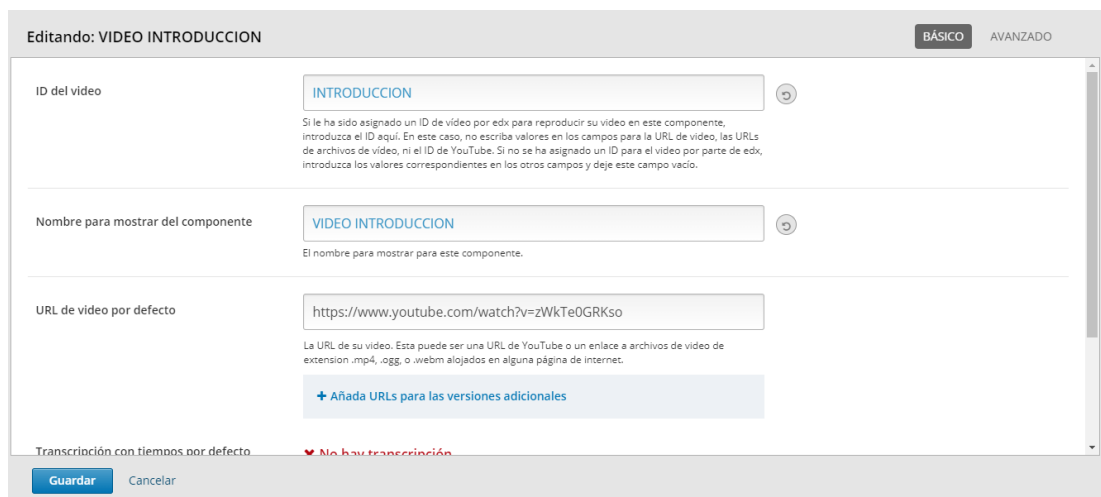


Figura 28. Configuraciones de Componente de Video en Unidad de Curso en plataforma OpenEDX

Presentaciones Power Point

Para ingresar una presentación se deberá subir la misma a la Herramienta de presentaciones, ir a Archivo y dar clic en publicar como se observa en la figura 29.

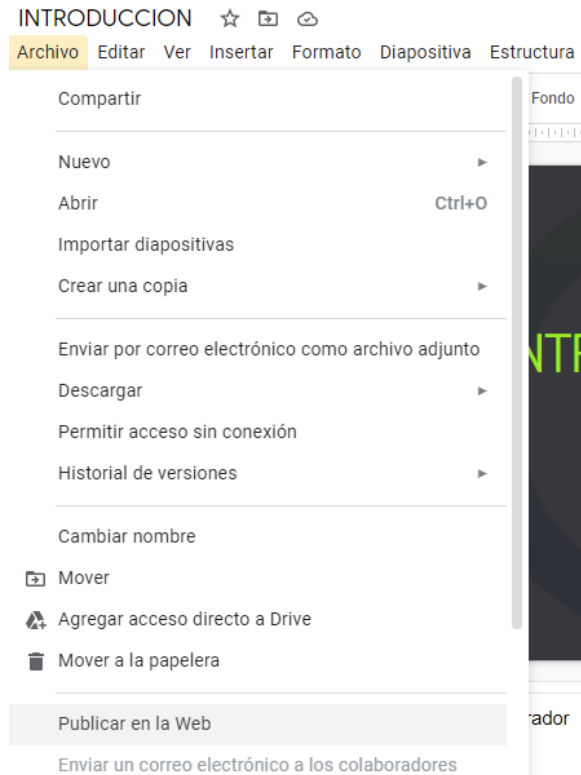


Figura 29. *Publicación Presentaciones en la Web Power Point*

A continuación se elige la opción “Incorporar” la cual se presenta en el cuadro presente en la figura 30 para copiar el enlace en formato HTML presente en el mismo.



Figura 30. *Incorporación de Presentación en formato HTML*

Se crea una nueva unidad como se observó anteriormente, y se añade un componente HTML. En la figura 31 se visualiza las opciones de documentación HTML que dispone la plataforma, elegimos la opción de Raw HTML

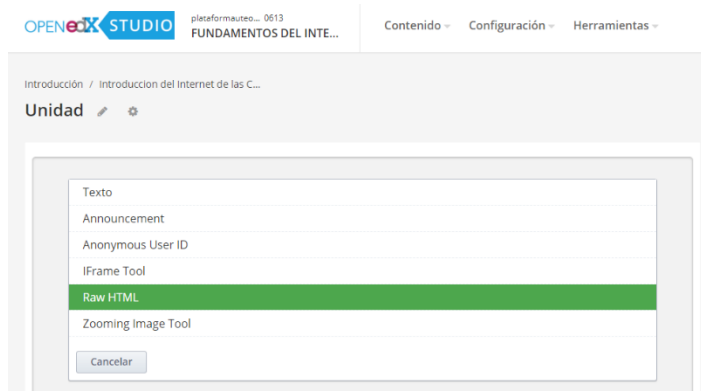


Figura 31. Plantillas HTML

A continuación se nos desplegará un cuadro donde podremos introducir cualquier tipo de texto en HTML. En la figura 32 se realiza la introducción del código HTML copiado de la presentación vista anteriormente.



Figura 32. Código HTML Presentación Power Point en OpenEDX

En la figura 33 se plasma el resultado final de la presentación embebida en el curso por la herramienta de OpenEDX



Figura 33. Presentación Power Point en OpenEDX.

Presentaciones Prezi

Para ingresar una presentación deberá estar contenida en una cuenta de Prezi, se da clic en opciones de presentación y luego en compartir enlace como se observa en la figura 34.

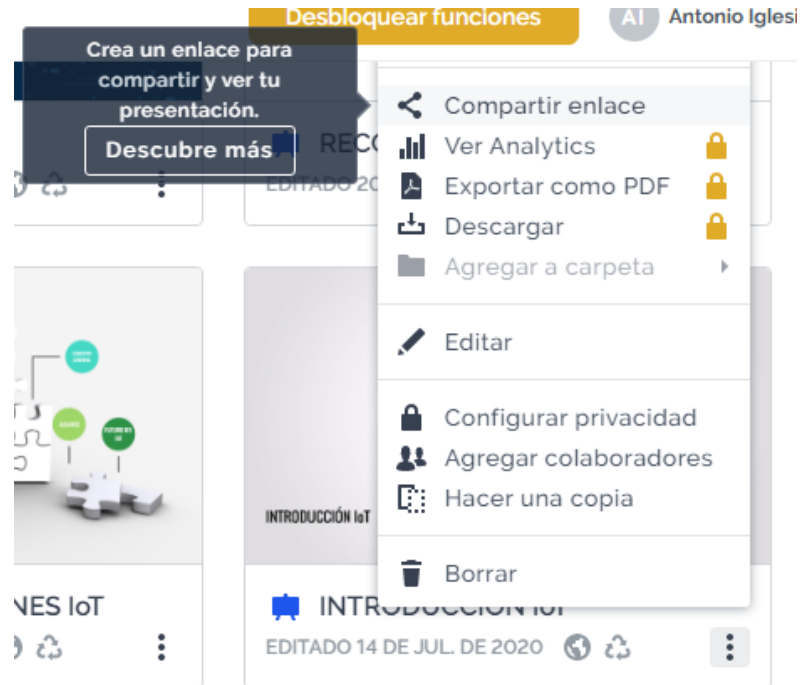


Figura 34. Publicación Presentaciones en la Web Prezi

A continuación se copia el enlace obtenido y se incorpora la presentación con una plantilla obtenida de la página oficial de Prezi en formato HTML la cual se presenta en la figura 35.

```
<iframe width="550" height="400" src="your_view_link/embed"
webkitallowfullscreen="1" mozallowfullscreen="1" allowfullscreen="1"></iframe>
```

Figura 35. Incorporación de Prezi en formato HTML

Se crea una nueva unidad como se observó anteriormente, y se añade un componente HTML. En la figura 36 se visualiza las opciones de documentación HTML que dispone la plataforma, elegimos la opción de Raw HTML

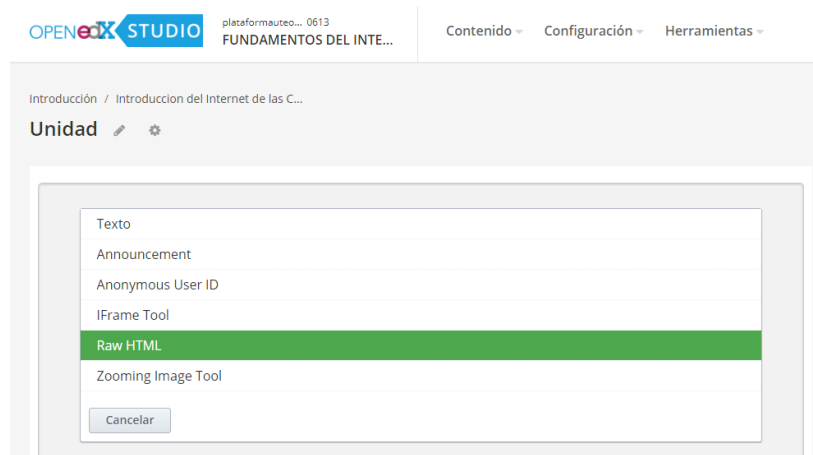


Figura 36. Plantillas HTML

A continuación se nos desplegará un cuadro donde podremos introducir cualquier tipo de texto en HTML. En la figura 37 se realiza la introducción del código HTML copiado de la presentación vista anteriormente.



Figura 37. Código HTML Presentación Prezi en OpenEDX

En la figura 38 se plasma el resultado final de la presentación embebida en el curso por la herramienta de OpenEDX



Figura 38. Presentación Prezi en OpenEDX

3.2.4. Fase 4: Evaluaciones y Canales de Comunicación Evaluaciones

En la tabla 14 se definen las tareas realizadas en el sprint correspondiente estableciendo las fechas de revisión en cada módulo.

Tabla 14. Fase 4 Evaluación

Fase 4. Tiempo: 15 días	Fecha de Cumplimiento: 15/04/2020 – 30/04/2020	
Tarea	Tiempo	Encargado
Preparación contenidos evaluaciones	3 días	Antonio Iglesias
Definición de canales de comunicación	3 días	Antonio Iglesias
Publicación Curso MOOC	9 días	Antonio Iglesias

Contenidos de Evaluaciones

Al ser un método de Evaluaciones Continuas, se ha determinado que sean evaluaciones de opción múltiple de 3 preguntas, de temas que se consideren claves para evaluar que el estudiante tenga una idea clara de los contenidos expuestos, además de comprobar que el estudiante puede proseguir con bases sólidas con respecto a lo expuesto en el curso, en la figura 39 se da un ejemplo de las preguntas planteadas en el curso MOOC

The image displays three sequential screenshots of a MOOC question interface. Each screenshot shows a question header, a point value, a question text, four radio button options, a 'Submit' button, and a 'Show Answer' button.

PREGUNTA No 1
 1 point possible (ungraded)
 El Internet de las cosas:

- Es un concepto que se refiere a la conexión del internet con el ambiente
- Es un concepto que se usa para referirse al traslado simple de datos
- Es un concepto que se refiere a una interconexión digital de objetos con el internet
- Es un concepto que se refiere a la vida que tienen los objetos gracias a internet

PREGUNTA No 2
 1 point possible (ungraded)
 La siglas IoT que representan globalmente al Internet de las cosas, en inglés significa:

- Internet of the Tools
- Internet of the Trade
- Internet of the Treasure
- Internet of the Things

PREGUNTA No 3
 1 point possible (ungraded)
 El concepto de internet indica que:

- Es un conjunto descentralizado de redes de comunicación interconectadas
- Es un conjunto centralizado de redes de comunicación interconectadas
- Es un conjunto centralizado de redes de comunicación interanexas
- Ninguna de las anteriores

Figura 39. Modelo Preguntas Curso MOOC.

Canales de comunicación

Se ha decidido que para efectuar una comunicación completa, se tendrá que exportar el curso a un entorno productivo, en el servicio Web de preferencia del Usuario, en nuestro caso hemos elegido google Cloud Plataform para albergar nuestro curso, y un gestor que administre la Base de datos en formato MySQL en la herramienta Web Service 000Webhost.

Publicación Curso MOOC

Migración Plataforma EdX

Una vez se haya concluido con la elaboración del curso en la plataforma OpenEDX en el entorno de desarrollo Debian, se hará la correspondiente migración al mismo:

- Se crea un directorio en donde contendrán todos los archivos para su migración como se indica en la figura 40. Luego se migrará las bases de MySQL y Mongo con los comandos detallados en las figuras 41 y 42

```
mkdir edx-migrate-data
```

Figura 40. Creación de Directorio de Migración EDX.

```
installdir/mysql/bin/mysqldump -u root -p --databases bitnami_edxapp  
bitnami_edxapp_csmh bitnami_xqueue --add-drop-database > edx-upgrade-  
data/bitnami_edx_mysql.sql
```

Figura 41. Migración Base de datos MySQL en EDX.

```
installdir/mongodb/bin/mongodump --authenticationDatabase admin -h localhost  
-u root -d bitnami_edxapp --out edx-upgrade-data/bitnami_edx_mongodb  
  
installdir/mongodb/bin/mongodump --authenticationDatabase admin -h localhost  
-u root -d bitnami_cs_comments_service --out edx-upgrade-  
data/bitnami_edx_mongodb
```

Figura 42. Migración Base de datos Mongo en EDX.

- Se copia las opciones de configuración definidas en la plataforma EdX y los archivos de datos (que contienen cargas, cursos, etc.). Los siguientes comandos detallados en la figura 43 también borran los activos para reducir el tamaño del directorio. Luego se copia los archivos de configuración de EdX XQueue y de EdX Forum como se detalla en las figuras 44 y 45 respectivamente, y finalmente se comprime todos estos datos para su exportación:

```
Cp installdir/apps/edx/conf/*.json edx-migrate-data  
Cp -R installdir/apps/edx/var edx-migrate-data  
Rm -rf edx-upgrade-data/var/staticfiles/*
```

Figura 43. Migración Archivos de datos en EDX.

```
cp installdir/apps/edx/conf/*.json edx-migrate-data
```

Figura 44. Migración Archivos .Json en EDX.

```
cp installdir/apps/forum/conf/httpd-forum.conf edx-migrate-data
```

Figura 45. Migración Base de datos Mongo en EDX.

- Luego se comprime todos estos datos que se encuentran contenidos en el directorio creado anteriormente con el comando que se detalla en la figura 46. Se tendrá que subir todos estos datos al servicio Web elegido, con el software FilleZilla, gracias a claves SSH. Se observa en la figuras 47 la subida de los datos al servicio virtualizado elegido, en este caso Google Plataform.

```
tar czf edx-migrate-data.tar.gz edx-migrate-data
```

Figura 46. Compresión Datos Plataforma EDX.

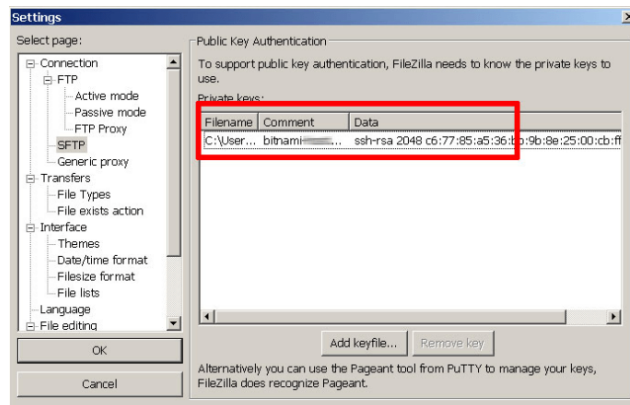


Figura 47. Software FileZilla.

- Luego en el servicio elegido mediante la consola simulada de entorno Debian se descomprime el archivo con el comando de la Figura No 48. Se realiza la importación de todos los servicios y datos contenidos en el archivo subido, se empieza por migrar la base de datos de tipo MySQL, con el comando detallado en la figura 49. Se migra la base de datos Mongo, con los comandos que se detalla en la figura 50.

```
tar xzf edx-migrate-data.tar.gz
```

Figura 48. Extracción de datos plataforma EDX.

```
installdir/mysql/bin/mysql -u root -p < edx-upgrade-data/bitnami_edx_mysql.sql
```

Figura 49. Migración Base de datos MySQL plataforma EDX.

```
installdir/mongodb/bin/mongorestore --authenticationDatabase admin -h localhost -u root --drop -d bitnami_edxapp edx-upgrade-data/bitnami_edx_mongodb/bitnami_edxapp
```

```
installdir/mongodb/bin/mongorestore --authenticationDatabase admin -h localhost -u root --drop -d bitnami_cs_comments_service edx-upgrade-data/bitnami_edx_mongodb/bitnami_cs_comments_service
```

Figura 50. Migración Base de datos MySQL plataforma EDX

- Se migra los datos de la plataforma EdX, reemplazando el directorio de datos de la plataforma EdX con el copiado de su instalación anterior de EdX con los comandos que se detalla en la figura 51. Se ejecuta migraciones y actualizaciones de la plataforma EdX con los comandos que se detalla en las figuras 52 y 53. Se migra los datos Edx Queue con los comandos que se detalla en la Figura 54. Se reinicia los servidores Apache, EdX Celery Workers y XQueue con los comandos que se detalla en la Figura 55

```
sudo rm -rf installdir/apps/edx/var
sudo cp -R edx-migrate-data/var installdir/apps/edx/var
sudo chown -R daemon:daemon installdir/apps/edx/var
```

Figura 51. Migración de datos plataforma EDX.

```
sudo installdir/apps/edx/bin/edxapp-migrate-lms
sudo installdir/apps/edx/bin/edxapp-migrate-cms
```

Figura 52. Migración LMS plataforma EDX.

```
sudo installdir/apps/edx/bin/edxapp-update-assets-lms
sudo installdir/apps/edx/bin/edxapp-update-assets-cms
```

Figura 53. Actualización LMS plataforma EDX.

```
sudo installdir/apps/xqueue/bin/django-admin.xqueue migrate --noinput
```

Figura 54. Migración datos Queue plataforma EDX.

```
sudo installdir/ctlscript.sh restart apache
sudo installdir/ctlscript.sh restart edx
sudo installdir/ctlscript.sh restart xqueue
```

Figura 55. Reinicio Servidores plataforma EDX

- Finalmente se verifica la integridad de los datos en la plataforma Google Open Plataforma como se ve en la figura 56. Además se observa en las figuras 57 y 58 la integridad del curso así como los recursos utilizados.

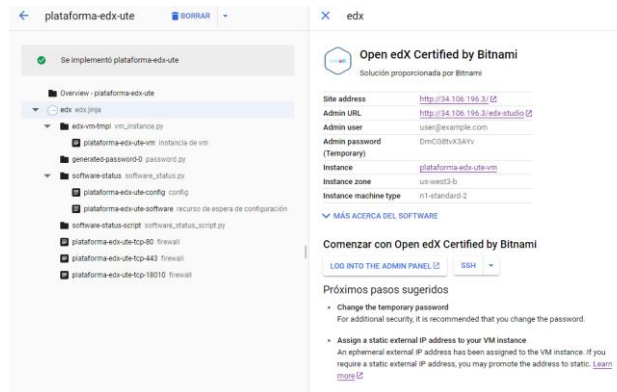


Figura 56. Google Open Plataforma OpenEDX

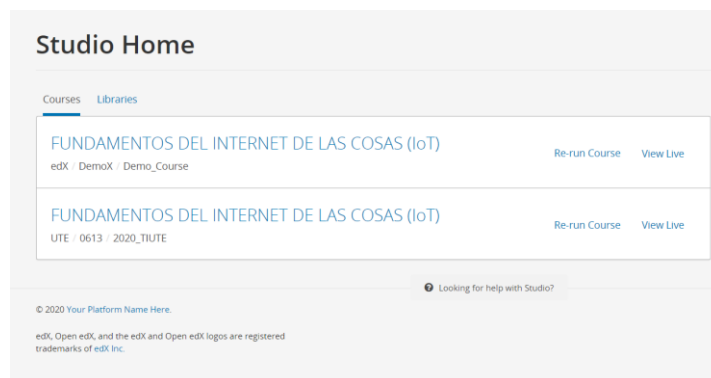


Figura 57. Cursos Plataforma EDX

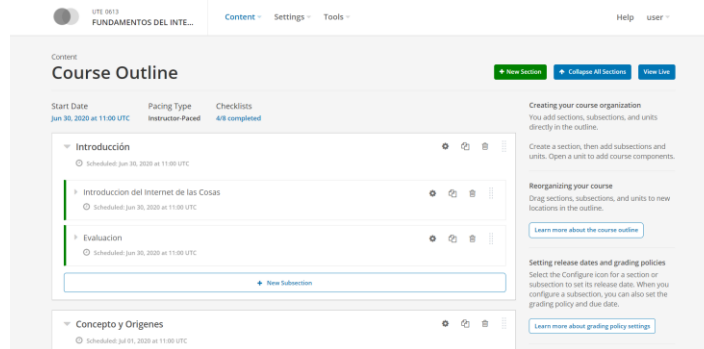


Figura 58. Curso “Fundamentos del Internet de las Cosas”

Base de Datos MySQL

Para tener acceso a una conexión de la base de datos Edx, se ha decidido subirla a un gestor gratuito de Base de datos para realizar la conexión con la aplicación anteriormente descrita, para esto debemos realizar los siguientes pasos descritos a continuación.

- Se genera un Script de toda la Base de datos generada en la plataforma de desarrollo OpenEDX alojada en Debian, para esto se arrancará el comando que se detalla en la Figura No 59 y se obtiene un archivo de extensión .sql como se puede observar en la figura 60

```
installdir/mysql/bin/mysql -u root -p < edx-upgrade-data/bitnami_edx_mysql.sql
```

Figura 59. Comando Obtención Script Base de Datos MySQL EDX

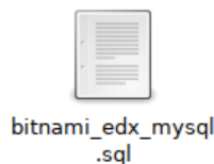


Figura 60. Script Base de Datos MySQL EDX

- Se ingresa al gestor de base de datos en la Web, y se ingresa a la configuración de la base de datos como se observa en la figura 61, luego se ejecuta el Script mencionado y se observa la integridad de los datos con sus respectivas tablas como se observa en la figura 62.

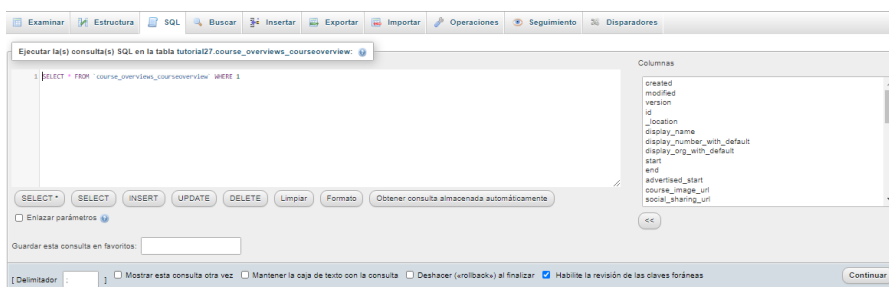


Figura 61. Ejecutor de Script Base de Datos MySQL

Mostrando filas 0 - 1 (total de 2. La consulta tardó 0.0015 segundos.) [display_name: [BLOB - 43 B]...] [BLOB - 24 B]...]

```
SELECT FROM course_overview_courseoverview ORDER BY display_name DESC
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: [Buscar en esta tabla] | Sort by key: Ninguna

Opciones

	created	modified	version	id	_location	display_name	display_number_with
<input type="checkbox"/>	2020-05-18 19:08:44.276597	2020-06-17 01:21:56.430012	0	course-v1:UTE+0813+2020_TIUITE	block-v1:UTE+0813+2020_TIUITE+type@course+block@cou...	FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)	0813
<input type="checkbox"/>	2020-05-17 21:49:51.844556	2020-05-17 21:49:56.907788	0	course-v1:edX+DemoX+Demo_Course	block-v1:edX+DemoX+Demo_Course+type@course+block@c...	edX Demonstration Course	DemoX

Seleccionar todo | Para los elementos que están marcados: [Editar] [Copiar] [Borrar] [Exportar]

Mostrar todo | Número de filas: 25 | Filtrar filas: [Buscar en esta tabla] | Sort by key: Ninguna

Figura 62. Tabla "course_overview_courseoverview" de Base de Datos MySQL OpenEDXOpenEDX

4. CONCLUSIONES Y RECOMENDACIONES

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- La implementación de un curso sobre fundamentos de IoT en una plataforma MOOC en conjunto con el desarrollo de una aplicación móvil conectada a la base de datos del mismo, permite la notificación de actividades y fechas importantes del mismo, por medio de notificaciones Push que se ejecutaran en actividades de segundo plano en el Sistema Operativo Android
- La realización, creación y curación de contenidos de “Fundamentos del Internet de las Cosas (IoT)” permitió el desarrollo de metodologías pedagógicas para elaborar la estructura de los mismos, así como su exposición en el curso.
- La instalación y configuración de la plataforma MOOC elegida llamada OpenEDX, permitieron la creación del curso MOOC “Fundamentos del Internet de las Cosas (IoT)” así como proveyeron de una base de Datos MySQL la cual se conectó con la aplicación móvil para proveer de las notificaciones deseadas
- La Implementación del curso “Fundamentos del Internet de las cosas IOT” utilizando una metodología de MOOC llamada xMOOC permite el uso de herramientas de código abierto para plasmar conocimientos adquiridos en un sitio didáctico de educación.
- La aplicación que genera notificaciones de actividades y fechas importantes del curso “Fundamentos de internet de las cosas (IoT)” permite al usuario manejar un mejor itinerario con respecto a las actividades que manejara siempre y cuando se inscriba a este curso.

4.2. RECOMENDACIONES

- Se recomienda el uso de plataformas MOOC de código abierto, así como herramientas Web para su almacenamiento en entornos productivos de uso gratuito, debido a que hay muchas herramientas en el mercado que cumplen este propósito, pero al alojar un curso de manera permanente las mismas tienen costos mensuales altos, por su capacidad de almacenamiento.
- Cuando se desea la promoción del curso, se recomienda impulsar la misma con la instalación de la aplicación desarrollada, esta deberá

estar disponible en un sitio de descarga libre, e instalación automática (tal como lo es Google Play), y dejar que el usuario reciba las notificaciones de Inscripción del curso, además de que simplifica el desarrollo de la misma.

- En el uso de plataformas MOOC, después de un análisis a las mismas, y a las metodologías globales que se han concretado en torno a MOOC, se recomienda el uso de la plataforma OpenEDX, debido a que la misma, en entornos de desarrollo Linux, tiene una capacidad de migración rápida e integra, además que su base de datos es fácilmente manipulable para su inserción en servicios Web que faciliten los datos y convertirlas en información útil para el administrador del mismo.

5. BIBLIOGRAFÍA

5. BIBLIOGRAFÍA

- Abrahamsson, Hanhineva, Hulkko, Ihme, Jääliñoja, Korkala, . . . Salo. (2004). *Mobile-D: an agile approach for mobile application development*. EEUU: ACM SIGPLAN conference on Object oriented programming systems, languages, and applications.
- Amaro, J. (2012). *El Gran libro de programación avanzada con android*. Barcelona: Marcombo.
- Amaya, Y. (2013). *Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles*. Tunja: Universidad Pedagógica y Tecnológica de Colombia.
- Android by Google. (18 de 5 de 2020). *Android Source*. Obtenido de Android Platform Codelab: <https://source.android.com/>
- Berrocoso, J. V. (2014). *MOOCS: UNA VISIÓN CRÍTICA DESDE LAS CIENCIAS DE LA EDUCACION*. Extremadura: Universidad de Extremadura.
- Blanco, P., Camarero, J., Fumero, A., Warterski, A., & Rodríguez, P. (2009). *Metodología de desarrollo ágil para sistemas móviles, Introducción al desarrollo con Android y el iPhone*. España: Universidad Politécnica de Madrid.
- Burton, F. (2012). *Android Application Development For Dummies*. EEUU: Willey.
- Cabero Almenara, J., Llorente Cejudo, M. d., & Vázquez Martínez, A. I. (2014). Las tipologías de MOOC: su diseño e implicaciones educativas. *Revista de Curriculum y Formación del Profesorado* , 13-26.
- Camarero, Fumero, Warterski, & Rodriguez. (2009). *Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhon*. Madrid: Doctorado en Ingeniería de Sistemas Telemáticos.
- Carina S. González, C. A. (2016). *Desafío en el diseño de MOOCs: incorporación de aspectos para*. Universidad de Murcia: Revista de Educación a Distancia.
- Carlos Castaño Garrido, I. M. (2015). Diseño, motivación y rendimiento en un curso MOOC cooperativo. *Comunicar*, 90.
- Dyba, & Dingsoyr. (2008). *Empirical Studies of Agile software Development*. EEUU: Information And Software Tecnology.
- Evans, D. (2011). *Internet de las cosas, Cómo la próxima evolución*. Estados Unidos: Cisco Internet Business Solutions Group (IBSG).

- Gómez, & Hernández. (2016). *MOBILE-D*. Quindio: Universidad de Quindio.
- Google Developers. (15 de 5 de 2020). *Android Developers*. Obtenido de Android Developers: <https://developer.android.com/guide/topics/media/media-formats>
- Greenough, J. (25 de Febrero de 2015). The corporate 'Internet of Things' will encompass more devices than the smartphone and tablet markets combined. *Bussines Insider*, pág. 1.
- HM Treasury and The Rt Hon George Osborne. (18 de Marzo de 2015). Budget 2015: some of the things we've announced. *Gov.UK*, pág. 1.
- Magrassi, Panarella, Deighton, & Johnson. (2011). Computers to Acquire Control of the Physical World. *Gartner research report*, 3.
- Mitchell, S., Villa, N., Stewart-Weeks, M., & Lange, A. (2013). The Internet of Everything for Cities. *Cisco Systems*, 1.
- Pernías, P., & Luján, S. (2013). *LOS MOOC: ORÍGENES, HISTORIA Y TIPOS*. España: Comunicación y Pedagogía, Universidad de Alicante.
- PORUP, J. (2016). "Internet of Things" security is hilariously broken and getting worse. *Ars Technica Addendum*, 1.
- RDFI. (1 de 10 de 2008). NFC Kit Launched to Spur Application Development. *RDFI Journal*, pág. 1.
- René Kizilcec, C. P. (2013). Deconstructing Disengagement: Analyzing Learner Subpopulations in Massive Open Online Courses. *Stanford Edu*, 1-10.
- Restivo, K. (16 de 05 de 2019). *Worldwide Quarterly Mobile Phone*. Obtenido de Worldwide Quarterly Mobile Phone: https://www.idc.com/tracker/showproductinfo.jsp?prod_id=37
- Schamer, O. (2015). MOOC 4.0: The Next Revolution in Learning & Leadership. *Huffpost*, 1.
- SCI FLO PORTUGAL. (2014). MOOC success factors: some critical considerations. *Revista Ibérica de Sistemas e Tecnologias de Informação*, 1.
- Seaton, L. B. (2012). Studying Learning in the Worldwide Classroom. *MIT*, 13.
- Sergio Luján Mora, P. P. (2013). Los MOOC: orígenes, historia y tipos. *Comunicación y Pedagogía*, 90.
- Shah, D. (2018). By The Numbers: MOOCS in 2017. *Class Central MOOC Report*, 1.
- Trakin Tech Network. (26 de Mayo de 2017). How IoT's Are Changing The Fundamentals Of "Retailing". *Trak.in*, pág. 1.

- Universidad Carlos III de Madrid. (Marzo de 2020). *Guía del profesor para la planificación, diseño e impartición de MOOCs*. Obtenido de http://docubib.uc3m.es/MOOCs/Guia-metodologica-MOOC-Wimba/page_02.htm
- Universidad de Alicante. (2012). *Departamento de Lenguajes y Sistemas Informáticos*. Obtenido de ¿Que son los cursos MOOC?: <http://desarrolloweb.dlsi.ua.es/cursos/2012/que-son-los-moocs/>
- Universitat Autònoma de Barcelona. (2013). *Diferencias Entre Cursos MOOC y Cursos Online*. Obtenido de <https://www.uab.cat/web/estudiar/mooc/diferencias-con-un-curso-online-1345668281279.html>
- UNIVERSITAT AUTÒNOMA DE BARCELONA. (1 de Marzo de 2020). *UNIVERSITAT AUTÒNOMA DE BARCELONA*. Obtenido de ¿Qué es un curso MOOC?: <https://www.uab.cat/web/estudiar/mooc/-que-es-un-curso-mooc-1345668281247.html>
- Vanegas, C. (2012). *VINCULOS - DESARROLLO DE APLICACIONES SOBRE ANDROID*. Caldas: Universidad Distrital Francisco José de Caldas.
- Vázquez, E., López, E., & Sarasola, J. (2013). *LA EXPANSIÓN DEL CONOCIMIENTO ABIERTO: LOS MOOC*. Barcelona: Octaedro.
- Zanjireh, M., & Larijani, H. (2015). A Survey on Centralised and Distributed Clustering Routing Algorithms for WSNs. *Research Gate*, 6.

6. ANEXOS

6. ANEXOS

Anexo 1. Código MainActivity

```
package com.example.openedxnotificacion;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;

import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.Reader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;

public class MainActivity extends AppCompatActivity {
    NotificationCompat.Builder notificacion;
    private static final int idunica = 123;
    //Button btnconsultar, btnGuardar;
    EditText etId, etCreated, etModified, etCreated1, etModified1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        notificacion = new NotificationCompat.Builder(this);
        setContentView(R.layout.activity_main);
        //btnconsultar = (Button)findViewById(R.id.btnConsultar);
        //btnGuardar = (Button)findViewById(R.id.btnGuardar);
        etId = (EditText)findViewById(R.id.etId);
        etCreated = (EditText)findViewById(R.id.etCreated);
        etModified = (EditText)findViewById(R.id.etModified);
        etCreated1 = (EditText)findViewById(R.id.etCreated1);
        etModified1 = (EditText)findViewById(R.id.etModified1);
        new
        ConsultarDatos().execute("http://archivosphpopenedx.000webhostapp.com/consult
a.php?display_name=FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)");
        new
        ConsultarDatos2().execute("http://archivosphpopenedx.000webhostapp.com/consul
ta2.php?display_name=FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)");
        hilo();
        //new
        ConsultarDatos().execute("http://10.0.3.2/CursoAndroid/consulta.php?id=2");
    }
}
```

```

        @Override
        public void onClick(View v) {

                //new
                ConsultarDatos().execute("http://10.0.3.2/CursoAndroid/consulta.php?id="+etId
                .getText().toString());
                new
                ConsultarDatos().execute("http://10.0.3.2/CursoAndroid/consulta.php?id=2");

        }
    });

    btnGuardar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

                new
                CargarDatos().execute("http://10.0.3.2/CursoAndroid/registro.php?nombres="+et
                Created.getText().toString()+"&tel="+etModified.getText().toString());

        }
    });*/

}

public void hilo(){
    try {
        Thread.sleep(60000);
        startService(new Intent(getApplicationContext(), MyService.class));
    } catch (InterruptedException e){
        e.printStackTrace();
    }
}

private class CargarDatos extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... urls) {

        // params comes from the execute() call: params[0] is the url.
        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may be invalid.";
        }
    }
    // onPostExecute displays the results of the AsyncTask.
    @Override
    protected void onPostExecute(String result) {

        Toast.makeText(getApplicationContext(), "Se almacenaron los datos
        correctamente", Toast.LENGTH_LONG).show();

    }
}
}

```



```

private class ConsultarDatos extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... urls) {
        // params comes from the execute() call: params[0] is the url.
        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may be invalid.";
        }
    }
    // onPostExecute displays the results of the AsyncTask.
    @Override
    protected void onPostExecute(String result) {
        JSONArray ja = null;
        try {
            ja = new JSONArray(result);
            etId.setText(ja.getString(0));
            etCreated.setText(ja.getString(1));
            etModified.setText(ja.getString(2));
            //etId.setText("FUNDAMENTOS DEL INTERNET DE LAS COSAS
(IoT)");
            //etCreated.setText("2020-07-10 17:00:00.000000");
            //etModified.setText("2020-07-16 23:30:00.000000");
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

```

private class ConsultarDatos2 extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... urls) {
        // params comes from the execute() call: params[0] is the url.
        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may be invalid.";
        }
    }
    // onPostExecute displays the results of the AsyncTask.
    @Override
    protected void onPostExecute(String result) {
        JSONArray ja = null;
        try {
            ja = new JSONArray(result);
            //etId.setText(ja.getString(0));
            etCreated1.setText(ja.getString(1));
            etModified1.setText(ja.getString(2));
            //etCreated1.setText("2020-06-27 11:00:00.000000");
            //etModified1.setText("2020-07-10 23:59:00.000000");
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

```

private String downloadUrl(String myurl) throws IOException {
    Log.i("URL", ""+myurl);
    myurl = myurl.replace(" ", "%20");
    InputStream is = null;
    // Only display the first 500 characters of the retrieved
    // web page content.
    int len = 500;

    try {
        URL = new URL(myurl);
        HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
        conn.setReadTimeout(10000 /* milliseconds */);
        conn.setConnectTimeout(15000 /* milliseconds */);
        conn.setRequestMethod("GET");
        conn.setDoInput(true);
        // Starts the query
        conn.connect();
        int response = conn.getResponseCode();
        Log.d("respuesta", "The response is: " + response);
        is = conn.getInputStream();

        // Convert the InputStream into a string
        String contentAsString = readIt(is, len);
        return contentAsString;

        // Makes sure that the InputStream is closed after the app is
        // finished using it.
    } finally {
        if (is != null) {
            is.close();
        }
    }
}

public String readIt(InputStream stream, int len) throws IOException,
UnsupportedEncodingException {
    Reader = null;
    reader = new InputStreamReader(stream, "UTF-8");
    char[] buffer = new char[len];
    reader.read(buffer);
    return new String(buffer);
}
}

```

Anexo 2. Código MyService

```
package com.example.openedxnotificacion;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

import android.app.AlarmManager;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;

import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.Reader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;

import java.net.MalformedURLException;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Timer;
import java.util.TimerTask;

import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.util.Log;
import android.widget.Toast;

public class MyService extends Service {
    NotificationCompat.Builder notificacion;
    private static final int idunica = 123;
    EditText etId, etCreated, etModified;

    //EditText etCreated, etModified;
```

```

@Override
    public void onCreate() {
        super.onCreate();
        mTimer = new Timer();
        mTimer.schedule(timerTask, 2000, 2 * 1000);
    }

    private Timer mTimer;

    TimerTask timerTask = new TimerTask() {
        @Override
        public void run() {
            Log.e("Log", "Running");
        }
    };

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {

        // TODO Auto-generated method stub
        //new
        MyService.ConsultarDatos().execute("http://10.0.3.2/CursoAndroid/consulta.php?id=2");
        new
        MyService.ConsultarDatos().execute("http://archivosphpopenedx.000webhostapp.com/consulta.php?display_name=FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)");
        new
        MyService.ConsultarDatos2().execute("http://archivosphpopenedx.000webhostapp.com/consulta2.php?display_name=FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)");
        return android.app.Service.START_STICKY;
    }

    public void hilo(){
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e){
            e.printStackTrace();
        }
    }

    public void ejecutar(){
        //new
        MyService.ConsultarDatos().execute("http://10.0.3.2/CursoAndroid/consulta.php?id=2");
        new
        MyService.ConsultarDatos().execute("http://archivosphpopenedx.000webhostapp.com/consulta.php?display_name=FUNDAMENTOS DEL INTERNET DE LAS

```

```

        // web page content.
        int len = 500;

        try {
            URL url = new URL(myurl);
            HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
            conn.setReadTimeout(10000 /* milliseconds */);
            conn.setConnectTimeout(15000 /* milliseconds */);
            conn.setRequestMethod("GET");
            conn.setDoInput(true);
            // Starts the query
            conn.connect();
            int response = conn.getResponseCode();
            Log.d("respuesta", "The response is: " + response);
            is = conn.getInputStream();

            // Convert the InputStream into a string
            String contentAsString = readIt(is, len);
            return contentAsString;

            // Makes sure that the InputStream is closed after the app
is
            // finished using it.
        } finally {
            if (is != null) {
                is.close();
            }
        }
    }

    public String readIt(InputStream stream, int len) throws
IOException, UnsupportedEncodingException {
        Reader reader = null;
        reader = new InputStreamReader(stream, "UTF-8");
        char[] buffer = new char[len];
        reader.read(buffer);
        return new String(buffer);
    }

    private class ConsultarDatos extends AsyncTask<String, Void,
String> {
        @Override
        protected String doInBackground(String... urls) {
            for (int i=1; i<=1;i++){
                hilo();
            }
            // params comes from the execute() call: params[0] is the
url.

            try {

```

```

        ja = new JSONArray(result);
        Date c = Calendar.getInstance().getTime();
        System.out.println("Current time => " + c);

        SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-
dd");

        String formattedDate = df.format(c);
        //etCreated.setText(ja.getString(1));
        //etModified.setText(ja.getString(2));

        String fechacreated = ja.getString(1).substring(0,10);
        String fechamodified =
ja.getString(2).substring(0,10);

        if(fechacreated.equals(formattedDate)){

notificacion.setSmallIcon(R.mipmap.ic_launcher);
                notificacion.setTicker("Nueva Notificacion");

notificacion.setWhen(System.currentTimeMillis());
                notificacion.setContentTitle("HOY INICIA EL
CURSO ");
                notificacion.setContentText("'FUNDAMENTOS DE
INTERNET DE LAS COSAS'");

                Intent intent = new Intent
(MyService.this,MyService.class);
                PendingIntent pendingintent =
PendingIntent.getActivity(MyService.this,0,intent,PendingIntent.FLAG_U
PDATE_CURRENT);
                notificacion.setContentIntent(pendingintent);
                NotificationManager nm = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
                nm.notify(idunica,notificacion.build());
        }else if(fechamodified.equals(formattedDate)){

notificacion.setSmallIcon(R.mipmap.ic_launcher);
                notificacion.setTicker("Nueva Notificacion2");

notificacion.setWhen(System.currentTimeMillis());
                notificacion.setContentTitle("HOY FINALIZA EL
CURSO");
                notificacion.setContentText("'FUNDAMENTOS DE
INTERNET DE LAS COSAS'");

                Intent intent = new Intent
(MyService.this,MyService.class);
                PendingIntent pendingintent =
PendingIntent.getActivity(MyService.this,0,intent,PendingIntent.FLAG_U

```

```

        for (int i=1; i<=1;i++){
            hilo();
        }
        // params comes from the execute() call: params[0] is the
url.
        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may be
invalid.";
        }
    }
    // onPostExecute displays the results of the AsyncTask.
    @Override
    protected void onPostExecute(String result) {
        notificacion = new
NotificationCompat.Builder(MyService.this);
        ejecutar();
        JSONArray ja = null;
        try {
            ja = new JSONArray(result);
            Date c = Calendar.getInstance().getTime();
            System.out.println("Current time => " + c);

            SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-
dd");

            String formattedDate = df.format(c);
            //etCreated.setText(ja.getString(1));
            //etModified.setText(ja.getString(2));

            String fechacreated1 =
ja.getString(1).substring(0,10);
            String fechamodified1 =
ja.getString(2).substring(0,10);

            if(fechacreated1.equals(formattedDate)){
                notificacion.setSmallIcon(R.mipmap.ic_launcher);
                notificacion.setTicker("Nueva Notificacion");
                notificacion.setWhen(System.currentTimeMillis());
                notificacion.setTitle("HOY ES EL DIA PARA
INSCRIBIRTE EN EL CURSO");
                notificacion.setText("'FUNDAMENTOS DE
INTERNET DE LAS COSAS'");

                Intent intent = new Intent
(MyService.this,MyService.class);
                PendingIntent pendingintent =
PendingIntent.getActivity(MyService.this,0,intent,PendingIntent.FLAG_U
PDATE_CURRENT);

```

```

(MyService.this, MyService.class);
        PendingIntent pendingintent =
PendingIntent.getActivity(MyService.this, 0, intent, PendingIntent.FLAG_U
PDATE_CURRENT);
        notificacion.setContentIntent(pendingintent);
        NotificationManager nm = (NotificationManager)
getService(NOTIFICATION_SERVICE);
        nm.notify(idunica, notificacion.build());
    }

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

@Override
public void onDestroy() {
    try {
        mTimer.cancel();
        timerTask.cancel();
    } catch (Exception e) {
        e.printStackTrace();
    }

    Intent intent = new Intent("com.android.openedxnotificacion");
    intent.putExtra("yourvalue", "torestore");
    sendBroadcast(intent);
}
}
}

```


Anexo 3. Código ReceiverCall

```
package com.example.openedxnotificacion;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.util.Log;

public class ReceiverCall extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        Log.i("Service Stops", "Ohhhhhhh");
        context.startService(new Intent(context, MyService.class));
    }
}
```

Anexo 4. Código SplashActivity

```
package com.example.openedxnotificacion;

import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.os.Handler;
import android.view.WindowManager;

public class splash_activity extends Activity {

    private final int DURACION_SPLASH = 3000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);

        setContentView(R.layout.activity_splash_activity);

        new Handler().postDelayed(new Runnable() {
            public void run() {
                Intent intent = new Intent(splash_activity.this,
                MainActivity.class);
                startActivity(intent);
                finish();
            }
        }, DURACION_SPLASH);
    }
}
```

Anexo 5. Código Activity_Main

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="20dp"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:paddingTop="20dp"
    android:background="@drawable/ute1"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="INFORMACION CURSO OPEN EDX"
        android:id="@+id/tvTitulo"/>

    <EditText
        android:id="@+id/etId"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tvTitulo"
        android:layout_marginTop="1dp"
        android:layout_marginBottom="10dp"
        android:hint="id" />

    <TextView
        android:id="@+id/tvTexto1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/etId"
        android:layout_marginTop="1dp"
        android:text="FECHA INICIO CURSO" />

    <EditText
        android:id="@+id/etCreated"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tvTexto1"
        android:layout_marginTop="1dp"
        android:layout_marginBottom="10dp"
        android:hint="created" />

    <TextView
        android:id="@+id/tvTexto2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/etCreated"
```

```

        android:hint="modified" />
<TextView
    android:id="@+id/tvTexto3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/etModified"
    android:layout_marginTop="1dp"
    android:text="FECHA INICIO INSCRIPCION DEL CURSO" />

<EditText
    android:id="@+id/etCreated1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvTexto3"
    android:layout_marginTop="1dp"
    android:layout_marginBottom="10dp"
    android:hint="created1" />

<TextView
    android:id="@+id/tvTexto4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/etCreated1"
    android:layout_marginTop="1dp"
    android:text="FECHA FIN INSCRIPCION DEL CURSO" />

<EditText
    android:id="@+id/etModified1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvTexto4"
    android:layout_marginTop="1dp"
    android:layout_marginBottom="10dp"
    android:hint="modified1" />

<!--<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Guardar"
    android:layout_below="@+id/etModified"
    android:id="@+id/btnGuardar"/>

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="consultar"
    android:layout_below="@+id/btnGuardar"
    android:id="@+id/btnConsultar"/>-->
</RelativeLayout>

```

Anexo 6. Código activity_splash_activity

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/ute"
    tools:context=".splash_activity">
```

Anexo 7. Splash Screen Aplicación Notificaciones Edx



CARGANDO, POR FAVOR ESPERE

FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)

UNIVERSIDAD UTE © TODOS LOS DERECHOS RESERVADOS

Anexo 8. Aplicación Notificaciones Edx

INFORMACION CURSO OPEN EDX 
**FUNDAMENTOS DEL INTERNET DE
LAS COSAS (IoT)**

FECHA INICIO CURSO

2020-08-14

FECHA FIN DE CURSO

2020-08-21

FECHA INICIO INSCRIPCION DEL CURSO

2020-08-10


FECHA FIN INSCRIPCION DEL CURSO

2020-08-13



UNIVERSIDAD UTE © TODOS LOS DERECHOS RESERVADOS

Anexo 9. Plataforma OpenEDX Studio Curso "FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)"


plataforma... 0613
FUNDAMENTOS DEL INTE...

Contenido Configuración Herramientas

Ayuda IglesiasA...

Contenido

Estructura del curso

+ Nueva Sección
Reindexar
+ Contratar todas las secciones
Ver en vivo

Start Date	Pacing Type	Checklists
Ago 14, 2020 a las 11:00 UTC	Instructor-Paced	4/9 completed

Introducción

Programado: Ago 14, 2020 a las 11:00 UTC

- ▶ **Introducción del Internet de las Cosas**
Programado: Ago 14, 2020 a las 11:00 UTC
- ▶ **Evaluación**
Programado: Ago 14, 2020 a las 11:00 UTC

+ Nueva Subsección

Concepto y Orígenes

Programado: Ago 14, 2020 a las 11:00 UTC

- ▶ **Concepto del Internet de las Cosas**
Programado: Ago 14, 2020 a las 11:00 UTC
- ▶ **Evaluación**
Programado: Ago 14, 2020 a las 11:00 UTC

+ Nueva Subsección

Características

Programado: Ago 14, 2020 a las 11:00 UTC

- ▶ **Características del Internet de las Cosas**
Programado: Ago 14, 2020 a las 11:00 UTC
- ▶ **Evaluación**
Programado: Ago 14, 2020 a las 11:00 UTC

+ Nueva Subsección

Recolección de la Información

Programado: Ago 14, 2020 a las 11:00 UTC

- ▶ **Recolección de la Información IoT**
Programado: Ago 14, 2020 a las 11:00 UTC
- ▶ **Evaluación**
Programado: Ago 14, 2020 a las 11:00 UTC

+ Nueva Subsección

Aplicaciones

Programado: Ago 14, 2020 a las 11:00 UTC

- ▶ **Aplicaciones Internet de las Cosas**
Programado: Ago 14, 2020 a las 11:00 UTC
- ▶ **Evaluación**
Programado: Ago 14, 2020 a las 11:00 UTC

+ Nueva Subsección

+ Nueva Sección

Crear la organización de su curso
Añada secciones, subsecciones y unidades directamente sobre la estructura del curso.

Crear una sección, luego añada subsecciones y unidades.
Abra una unidad para añadirle componentes de curso.

Reorganice su curso
Arrastre secciones, subsecciones y unidades a nuevas posiciones en la estructura del curso.

Aprenda más sobre la estructura del curso

Configurar fechas de liberación y políticas de calificaciones
Seleccione el icono de configuración para una sección o subsección para definir su fecha de liberación. Al configurar una subsección, también puede definir la política de calificación y la fecha límite de entrega.

Aprender más sobre la configuración de política de calificaciones.

Cambiar el contenido que los estudiantes pueden ver
Para publicar un borrador del contenido, seleccione el icono de Publicar de la respectiva sección, subsección o unidad.

Para volver disponible una sección, subsección o unidad para los estudiantes, seleccione el icono de Configuración para ese nivel, a continuación, seleccione la **opción apropiada**. Las calificaciones obtenidas en secciones, subsecciones y unidades ocultas no se incluyen en los cálculos de calificaciones.

Para ocultar el contenido de una subsección a los estudiantes después de que la fecha límite de la subsección ha pasado, seleccione el icono de Configuración y a continuación, seleccione **Ocultar contenido después de la fecha límite**. Las calificaciones obtenidas en este caso si serán tenidas en cuenta en los cálculos de calificaciones.

Aprender más sobre la configuración de visibilidad de contenidos

¿Buscando ayuda con Studio?

© 2020 Open edX | Powered by eduNEXT.

edX, Open edX, y los logos de edX y Open edX son marcas registradas de edX Inc.

Español (Latinoamérica)

POWERED BY
OPENEDX

Anexo 10. Plataforma OpenEDX LMS Curso "FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)"

The screenshot shows the OpenEDX LMS interface for the course 'FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)'. At the top, the UTE logo is visible on the left, and the course ID 'plataformauteopenedx:0613' and course name are in the center. A user profile 'IglesiasAntonio1996' is shown on the right. Below the header, there is a navigation bar with 'Curso', 'Discusión', 'Progreso', 'Notas', and 'Instructor'. The main content area features the course title, a search bar, and a 'Buscar' button. A notification box states: 'El curso empieza en 3 semanas el 14 de ago. de 2020. No olvide añadir un recordatorio a su calendario!' with an 'Expandir todo' button. A list of course sections is provided: 'Introducción', 'Concepto y Origenes', 'Características', 'Recolección de la Información', and 'Aplicaciones'. On the right, 'Herramientas del Curso' includes 'Marcadores', and 'Fechas importantes del curso' lists the start date (Aug 14, 2020) and end date (Aug 21, 2020). The footer contains the OpenEDX logo, text about the platform, and links for 'Sobre eduNEXT', 'Publique sus cursos en Open edX', 'Términos del servicio', and 'Política de privacidad'.

plataformauteopenedx:0613
FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT) Cursos IglesiasAntonio1996

Ver este curso como: Equipo

Curso Discusión Progreso Notas Instructor

FUNDAMENTOS DEL INTERNET DE LAS COSAS (IoT)

Buscar en el curso

El curso empieza en 3 semanas el 14 de ago. de 2020.
No olvide añadir un recordatorio a su calendario!

- > **Introducción**
- > **Concepto y Origenes**
- > **Características**
- > **Recolección de la Información**
- > **Aplicaciones**

Herramientas del Curso

- Marcadores**

Fechas importantes del curso

Hoy es Jul 27, 2020 00:31 -05

Inicio del Curso
Dentro de 3 semanas - Aug 14, 2020

Finalización del curso
Dentro de 1 mes - Aug 21, 2020

Después de esta fecha, el contenido del curso será archivado.

Cursos en línea de plataformauteopenedx. Este sitio de Open edX está construido con los servicios en la nube de eduNEXT

[Sobre eduNEXT](#) [Publique sus cursos en Open edX](#)

[Términos del servicio](#) [Política de privacidad](#)

Derechos reservados - 2019