

Guía Pedagógica y de Evaluación del Módulo Diseño y desarrollo de aplicaciones móviles

I. Guía Pedagógica del Módulo Diseño y desarrollo de aplicaciones móviles

Editor: Colegio Nacional de Educación Profesional Técnica

Guía pedagógica y de evaluación del Módulo: Diseño y desarrollo de aplicaciones móviles.

Área(s): Tecnología y transporte.

Carrera(s): Informática

Semestre(s): Sexto

© Colegio Nacional de Educación Profesional Técnica

Fecha de diseño o actualización: Junio de 2017.

Vigencia: Dos años, en tanto no se produzca un documento que lo anule o desaparezca el objeto del actual.

Prohibida la reproducción total o parcial de esta obra por cualquier medio, sin autorización por escrito del Conalep.

Directorio

Directora General
Candita Victoria Gil Jiménez

Secretario General
Roger Armando Frías Frías

Secretaria Académica
María Elena Salazar Peña

Secretaria de Administración
Corazón de María Madrigal

Secretario de Planeación y Desarrollo Institucional
Francisco Cuauhtémoc Santiago Jaime

Secretario de Servicios Institucionales
Javier Rodrigo Villegas Garcés

Director Corporativo de Asuntos Jurídicos
Juan Carlos Castillo Guzmán

Titular de la Unidad de Estudios e Intercambio Académico
Patricia Guadalupe Guadarrama Hernández

Director Corporativo de Tecnologías Aplicadas
Humberto Zentella Falcón

Directora de Diseño Curricular
Silvia Alejandra Guzmán Saldaña

Coordinadora de las Áreas Básicas y de Servicios
Caridad del Carmen Cruz López

Coordinador de las Áreas de Mantenimiento e Instalación,
Electricidad, Electrónica y TIC
Marco Antonio Valadez Pérez

Coordinadora de las Áreas de Procesos de Producción y
Transformación
Norma Osorio Vera

Grupo de trabajo:

Diseño:

Técnico:

Con la asesoría de:
Fundación Internacional para la Juventud IYF.
Programa Clave.

Metodológico:

Con la asesoría de:
Fundación Internacional para la Juventud IYF.
Programa Clave.

Contenido

	Pág.
I. Guía pedagógica	
❖ Descripción	6
❖ Datos de identificación del estándar de competencia	7
❖ Generalidades pedagógicas	8
❖ Enfoque del módulo	10
❖ Orientaciones didácticas y estrategias de aprendizaje por unidad	11
❖ Prácticas/Actividades	17
II. Guía de evaluación	40
❖ Descripción	41
❖ Tabla de ponderación	44
❖ Desarrollo de actividades de evaluación	46
❖ Matriz de valoración o rúbrica	53

1. Descripción

La Guía Pedagógica es un documento que integra elementos técnico-metodológicos planteados de acuerdo con los principios y lineamientos del **Modelo Académico del CONALEP** para orientar la práctica educativa del docente en el desarrollo de competencias previstas en los programas de estudio.

La finalidad que tiene esta guía es facilitar el aprendizaje de los alumnos, encauzar sus acciones y reflexiones y proporcionar situaciones en las que desarrollará las competencias. El docente debe asumir conscientemente un rol que facilite el proceso de aprendizaje, proponiendo y cuidando un encuadre que favorezca un ambiente seguro en el que los alumnos puedan aprender, tomar riesgos, equivocarse extrayendo de sus errores lecciones significativas, apoyarse mutuamente, establecer relaciones positivas y de confianza, crear relaciones significativas con adultos a quienes respetan no por su estatus como tal, sino como personas cuyo ejemplo, cercanía y apoyo emocional es valioso.

Es necesario destacar que el desarrollo de la competencia se concreta en el aula, ya que **formar con un enfoque en competencias significa crear experiencias de aprendizaje para que los alumnos adquieran la capacidad de movilizar, de forma integral, recursos que se consideran indispensables para saber resolver problemas en diversas situaciones o contextos**, e involucran las dimensiones cognitiva, afectiva y psicomotora; por ello, los programas de estudio, describen las competencias a desarrollar, entendiéndolas como la combinación integrada de conocimientos, habilidades, actitudes y valores que permiten el logro de un desempeño eficiente, autónomo, flexible y responsable del individuo en situaciones específicas y en un contexto dado. En consecuencia, la competencia implica la comprensión y transferencia de los conocimientos a situaciones de la vida real; ello exige relacionar, integrar, interpretar, inventar, aplicar y transferir los saberes a la resolución de problemas. Esto significa que **el contenido, los medios de enseñanza, las estrategias de aprendizaje, las formas de organización de la clase y la evaluación se estructuran en función de la competencia a formar**; es decir, el énfasis en la proyección curricular está en lo que los alumnos tienen que aprender, en las formas en cómo lo hacen y en su aplicación a situaciones de la vida cotidiana y profesional.

Considerando que el alumno está en el centro del proceso formativo, se busca acercarle elementos de apoyo que le muestren qué **competencias** va a desarrollar, cómo hacerlo y la forma en que se le evaluará. Es decir, mediante la guía pedagógica el alumno podrá **autogestionar su aprendizaje** a través del uso de estrategias flexibles y apropiadas que se transfieran y adopten a nuevas situaciones y contextos e ir dando seguimiento a sus avances a través de una autoevaluación constante, como base para mejorar en el logro y desarrollo de las competencias indispensables para un crecimiento académico y personal.

2. Datos de Identificación del estándar de competencia

Título:			
Código:		Nivel de competencia:	
Elementos de competencia laboral			

3. Generalidades pedagógicas

El docente debe asumir conscientemente un rol que facilite el proceso de aprendizaje, proponiendo y cuidando un encuadre que favorezca un ambiente seguro en el que los alumnos puedan aprender, apoyarse mutuamente y establecer relaciones positivas y de confianza.

Por otro lado, considerando que el alumno está en el centro del proceso formativo, se busca acercarle elementos de apoyo que le muestren qué competencias va a desarrollar, cómo hacerlo y la forma en que se le evaluará. Es decir, mediante la guía pedagógica el alumno podrá autogestionar su aprendizaje a través del uso de estrategias flexibles y apropiadas que se transfieran y adapten a nuevas situaciones y contextos e ir dando seguimiento a sus avances a través de una autoevaluación constante, como base para mejorar en el logro y desarrollo de las competencias indispensables para un crecimiento académico y personal.

Con el propósito de difundir los criterios a considerar en la instrumentación de la presente guía entre los docentes y personal académico de planteles y Colegios Estatales, se describen algunas consideraciones respecto al desarrollo e intención de las competencias expresadas en los módulos correspondientes a la formación básica, propedéutica y profesional.

Los principios asociados a la concepción constructivista del aprendizaje mantienen una estrecha relación con los de la educación basada en competencias, la cual se ha concebido en el Colegio como el enfoque idóneo para orientar la formación ocupacional de los futuros profesionales técnicos y profesional técnicos-bachiller. Este enfoque constituye una de las opciones más viables para lograr la vinculación entre la educación y el sector productivo de bienes y servicios.

En este sentido, se debe considerar que el papel que juegan el alumno y el docente en el marco del Modelo Académico del CONALEP tenga, entre otras, las siguientes características:

El alumno:	El docente:
<ul style="list-style-type: none"> ❖ Mejora su capacidad para resolver problemas. ❖ Aprende a trabajar en grupo y comunica sus ideas. ❖ Aprende a buscar información y a procesarla. ❖ Construye su conocimiento. ❖ Adopta una posición crítica y autónoma. ❖ Realiza los procesos de autoevaluación y coevaluación. 	<ul style="list-style-type: none"> ❖ Organiza su formación continua a lo largo de su trayectoria profesional. ❖ Domina y estructura los saberes para facilitar experiencias de aprendizaje significativo. ❖ Planifica los procesos de enseñanza y de aprendizaje atendiendo al enfoque por competencias, y los ubica en contextos disciplinares, curriculares y sociales amplios. ❖ Lleva a la práctica procesos de enseñanza y de aprendizaje de manera efectiva, creativa e innovadora a su contexto institucional. ❖ Evalúa los procesos de enseñanza y de aprendizaje con un enfoque formativo. ❖ Construye ambientes para el aprendizaje autónomo y colaborativo. ❖ Contribuye a la generación de un ambiente que facilite el desarrollo sano e integral de los estudiantes. ❖ Participa en los proyectos de mejora continua de su escuela y apoya la gestión institucional.

El docente en lugar de transmitir vertical y unidireccionalmente los conocimientos, es un mediador del aprendizaje, ya que:

- Planea y diseña experiencias y actividades necesarias para la adquisición de las competencias previstas. Asimismo, define los ambientes de aprendizaje, espacios y recursos adecuados para su logro.
- Proporciona oportunidades de aprendizaje a los estudiantes apoyándose en metodologías y estrategias didácticas pertinentes a los Resultados de Aprendizaje.
- Ayuda también al alumno a asumir un rol más comprometido con su propio proceso, invitándole a tomar decisiones.
- Facilita el aprender a pensar, fomentando un nivel más profundo de conocimiento.
- Ayuda en la creación y desarrollo de grupos colaborativos entre los alumnos.
- Guía permanentemente a los alumnos.
- Motiva al alumno a poner en práctica sus ideas, animándole en sus exploraciones y proyectos.

4. Enfoque del Módulo

El módulo de **Diseño y desarrollo de aplicaciones móviles** introduce al alumno en los componentes básicos de la metodología SCRUM para el desarrollo de aplicaciones bajo el modelo de programación ágil para equipos de trabajo, lo que permite controlar el proceso de calidad de desarrollo de software. Del mismo modo, incluye las bases para manejar la plataforma de desarrollo Java con el conjunto de librerías que permiten crear aplicaciones particulares para dispositivos Android. Además permite que el alumno utilice las estructuras de programación para la instauración de actividades e interfaces de usuario para el desarrollo típico de una aplicación de Android, así como estructuras de programación que permiten el despliegue de contenidos variables. Finalmente, involucra al alumno en la aplicación de estructuras para programación e interactuar con elementos multimedia y servicios web, así como con dispositivos externos para crear aplicaciones de Internet de las Cosas.

Además, estas competencias se complementan con la incorporación de otras competencias básicas, las profesionales y genéricas que refuerzan la formación tecnológica y científica, y fortalecen la formación integral de los educandos; que los prepara para comprender los procesos productivos en los que está involucrado para enriquecerlos, transformarlos, resolver problemas, ejercer la toma de decisiones y desempeñarse en diferentes ambientes laborales, con una actitud creadora, crítica, responsable y propositiva; de la misma manera, fomenta el trabajo en equipo, el desarrollo pleno de su potencial en los ámbitos profesional y personal y la convivencia de manera armónica con el medio ambiente y la sociedad.

La tarea docente en este módulo tendrá que diversificarse, a fin de que los Docentes realicen funciones preceptoras, las que consistirán en la guía y acompañamiento de los alumnos durante su proceso de formación académica y personal y en la definición de estrategias de participación que permitan incorporar a su familia en un esquema de corresponsabilidad que coadyuve a su desarrollo integral; por tal motivo, deberá destinar tiempo dentro de cada unidad para brindar este apoyo a la labor educativa de acuerdo al Programa de Preceptorías.

Por último, es necesario que al final de cada unidad de aprendizaje se considere una sesión de clase en la cual se realice la recapitulación de los aprendizajes logrados, en lo general, por los alumnos, con el propósito de verificar que éstos se han alcanzado o, en caso contrario, determinar las acciones de mejora pertinentes. En este proceso, los docentes tienen la facultad de instrumentar las modalidades de autoevaluación, coevaluación y heteroevaluación, de acuerdo con las condiciones particulares de su entorno, aun cuando de manera institucional se definen los criterios e indicadores para su aplicación.

5. Orientaciones didácticas y estrategias de aprendizaje por unidad

Unidad I	Administración del ciclo de desarrollo de aplicaciones
Orientaciones Didácticas	

Para el desarrollo de la presente unidad se recomienda al docente:

- Aplicar una evaluación inicial al comenzar la unidad, definiendo los contenidos y resultados de aprendizaje a alcanzar, también se exhorta a establecer reglas grupales para el desarrollo de las clases y acuerdos en el cumplimiento de las tareas encomendadas dentro y fuera del aula, y por último es relevante promover la asistencia y participación diaria, el cuidado de los materiales, recursos didácticos y equipo de trabajo.
- Presentar el tema “Identificación de componentes de un app” por medio de una dinámica dirigida en el laboratorio de cómputo, utilizando herramientas de desarrollo como Eclipse o Android Studio.
- Solicitar investigación documental sobre las herramientas no utilizadas en clase.
- Desarrollar los ejercicios prácticos sobre los componentes de un app: clases y layouts.
- Elaborar prácticas con dispositivos físicos, comentando sobre la variedad de dispositivos fabricados en el mercado y adoptados por el consumo de la región.
- Presentar los temas de “diseño de layout” y “diseño de controles” por medio de recursos interactivos y ejemplos de programas.
- Solicitar los ejercicios de programación que requieran el uso de estructuras de programación Java aprendidas en cursos pasados: secuencia, ciclos y selección.
- Solicitar un mapa mental comparando el modelo de desarrollo de cascada con el modelo de desarrollo ágil.
- Organizar una lluvia de ideas sobre las ventajas y desventajas de utilizar una metodología ágil de desarrollo.
- Exponer frente a grupo los elementos básicos del lenguaje: variables y tipos de datos.
- Desarrollar una exposición utilizando multimedia u otros recursos para presentar a los actores y componentes de la metodología SCRUM.
- Solicitar una investigación documental de empresas mexicanas que utilicen SCRUM en sus procesos de desarrollo.
- Promover la competencia de administración de proyectos de TI con SCRUM por medio de un juego: viaje a Marte, Ciudad de Legos, Barquitos de Papel, u otra, simulando la aplicación de la metodología SCRUM para su desarrollo.

- Subraya la importancia que tiene la presencia del alumno en cada clase, su participación para el enriquecimiento del aprendizaje de todo el grupo y la asignación de tareas y actividades intra y extramuros, con el fin de incentivar en él su cumplimiento voluntario y oportuno.
- Organiza sistemáticamente la información que se ha de manejar y procesar para su aprendizaje. Efectuando explícitamente la vinculación de esta unidad con la que precede, con el fin de que el alumno valore su importancia académica y curricular.
- Promueve la dinámica grupal colaborativa y cooperativa a través de la realización de las técnicas didácticas y de aprendizaje correspondientes, durante el transcurso de cada sesión para favorecer el clima que fomente el intercambio constructivo de ideas.

Estrategias de Aprendizaje	Recursos didácticos
<ul style="list-style-type: none"> • Realizar de manera individual una investigación documental sobre la metodología de cascada para el desarrollo de software • Hacer un cuadro sinóptico con las diferentes versiones del sistema operativo Android y la implicación de esta diversidad en el desarrollo de aplicaciones móviles • Explorar, junto con el profesor, la herramienta visual para el desarrollo de aplicaciones. Discute su relevancia en el entorno laboral de tu región. • Realizar la práctica núm.1 Diseño de primera aplicación “Hola Mundo” • Realizar la práctica núm.2 “Manejo de evento de calculadora numérica”. • Desarrolla en el laboratorio de cómputo los ejercicios sugeridos por el profesor. Consulta bibliografía del lenguaje Java para acceder a sintaxis con la que no estés familiarizado • Realizar la actividad de evaluación 1.1.1 • Revisa con el profesor los componentes del modelo SCRUM • Realizar la actividad núm. 3 “Simulación de jornada SCRUM”. • Realizar la práctica núm. 4 “Desarrollo de un app con SCRUM”. • Realizar la actividad de evaluación 1.2.1 	<p>Mednieks, Z. et al. (2011). <i>Programming Android</i>. Estados Unidos de América, O’Reilly.</p> <p>Cotton, J. (2012). <i>Android Programming for Beginners</i>. Estados Unidos de América, O’Reilly.</p> <p>Stroud, A. y Milette, G. (2012). <i>Professional Android Sensor Programming</i>. Estados Unidos de América, AbeBooks.</p> <p>Letelier, P. (2011). <i>Actividad: Lo básico de Kanban y Scrum construyendo una Lego City</i>, Blogspot. Recuperado el 29/01/17 de http://agilismoatwork.blogspot.mx/2011/11/actividad-para-aprender-lo-basico-de.html</p> <p>Albaladejo, X. (2009). El expendedor – Juego de simulación de Scrum. Proyectosagiles. Recuperado el 29/01/17 de https://proyectosagiles.org/2009/09/13/expendedor-juego-simulacion-scrum/</p>

Unidad II	Programación de la interfaz de la aplicación
Orientaciones Didácticas	

Para el desarrollo de la presente unidad se recomienda al docente:

- Desarrollar una actividad en el laboratorio de cómputo para la comprensión de los elementos de Layout XML y Clase Java que componen una actividad, fomentando en todo momento el aprendizaje por cuenta propia.
- Demostrar de manera interactiva en el laboratorio de cómputo la manera de integrar controles gráficos a una actividad utilizando una GUI y utilizando código.
- Exponer frente al grupo el uso de la clase Intent para la interacción entre Actividades.
- Solicitar un cuadro sinóptico con los atributos de la clase Intent y su uso para paso de parámetros entre Actividades.
- Ejemplificar por medio de demostraciones interactivas la creación e implementación de respuesta a eventos iniciados por el usuario.
- Solicitar un mapa mental de los diferentes sensores disponibles en un dispositivo móvil y los eventos en Java para capturarlos y responder a ellos.
- Solicitar la resolución de ejercicios que involucren la navegación entre actividades, paso de parámetros y uso avanzado del lenguaje Java.
- Presentar con demostraciones interactivas y contenidos multimedia uno de los constructos en Java para presentar datos repetitivos de longitud variable. Selecciona entre GridView, ListView o Recycler View.
- Introducir por medio de un ejercicio guiado, el concepto de Fragmento y su relación con los contenedores variables.
- Establecer por medio de ejercicios realizados en dúos o equipos, el concepto de Adaptadores.
- Solicitar una investigación bibliográfica documentada en ensayo o cuadro sinóptico sobre conceptos de ciclos y manejo de arreglos en su relación con el llenado de fragmentos.
- Solicitar un mapa mental o cuadro sinóptico con los constructos de Java para datos variables no vistos en clase.
- Solicitar el desarrollo en equipo de una aplicación con datos repetitivos utilizando los tres constructos.
- Organizar sistemáticamente la información que se ha de manejar y procesar para su aprendizaje. Efectuando explícitamente la vinculación de esta unidad con la que precede, con el fin de que el alumno valore su importancia académica y curricular.
- Promover la dinámica grupal colaborativa y cooperativa a través de la realización de las técnicas didácticas y de aprendizaje correspondientes, durante el transcurso de cada sesión para favorecer el clima que fomente el intercambio constructivo de ideas.
- Presentar sus tareas en forma ordenada, busca soluciones a los problemas que le presentan y cumple con las actividades que planifica. Sigue instrucciones y procedimientos de manera reflexiva en la realización de cada una de las prácticas realizadas.

Estrategias de Aprendizaje	Recursos Académicos
<ul style="list-style-type: none"> • Revisar de manera individual los recursos sugeridos sobre el uso y creación de Layouts. • Realizar las actividades guiadas por el instructor para la creación de Layouts • Construir un cuadro sinóptico con los métodos y atributos de la clase Intent • Realizar la práctica núm. 5 “Creación ventanas de actividades interconectadas”. • Revisar de manera individual los recursos sugeridos para conocer el acceso a los sensores del móvil. • Realizar la práctica núm.6 “Modificación de imágenes” • Realizar la práctica núm. 7 “Detección de acelerómetro en el dispositivo”. • Realizar la actividad de evaluación 2.1.1 • Realizar las actividades guiadas por el instructor para la creación de contenidos variables y Fragmentos • Realizar la practica núm. 8 “Despliegue de fragmentos”. • Realizar un mapa mental que compare los diferentes layouts para presentar contenido variable. • Realizar la actividad de evaluación 2.2.1 	<p>Burd, B. (2012). Android Application Development for Dummies. Estados Unidos de América, Goodreads.</p> <p>Software</p> <ul style="list-style-type: none"> • Software Eclipse Mars o superior, o Android Studio 2.0 o superior • http://stackoverflow.com/questions/14574879/how-to-detect-movement-of-an-android-device • http://stackoverflow.com/questions/5271448/how-to-detect-shake-event-with-android • https://code.tutsplus.com/tutorials/using-the-accelerometer-on-android--mobile-22125 • https://developer.android.com/guide/topics/ui/declaring-layout.html • Recycler Views. Recuperado el 15/12/16 de https://www.youtube.com/watch?v=8ePqYGMxdSY

Unidad III	Programación de interacciones con datos externos
Orientaciones Didácticas	


Para el desarrollo de la presente unidad se recomienda al docente:

- Enfatizar los objetivos del módulo y su importancia dentro del ecosistema laboral.
- Utilizar el laboratorio de cómputo para demostrar los métodos de acceso a recursos multimedia en la estructura de archivos de una aplicación, usando Android Studio o Eclipse.
- Solicitar el desarrollo de ejercicios en duplas para la creación de Actividades que consuman recursos internos.
- Realizar las prácticas de laboratorio para demostrar la interacción de una aplicación con otras aplicaciones.
- Demostrar por medio de videos o contenido multimedia los métodos para lograr animaciones simples con objetos multimedia almacenados de manera local.
- Solicitar una infografía con aplicaciones reales que se apoyen en el uso de recursos almacenados de manera interna.
- Guiar una práctica para introducir la clase auxiliar para el acceso a bases de datos SQLite.
- Explicar con apoyo de medios electrónicos los conceptos de web service y flujos (streams) de datos.
- Solicitar un mapa mental para describir los elementos de los estándares de transmisión de información XML y json.
- Describir en un ejercicio práctico las clases de Java para interpretar datos en formatos XML y json.
- Solicitar una infografía que describa el concepto de Internet de las Cosas (IoT).
- Utilizar el laboratorio de cómputo para desarrollar en equipo una aplicación que se comunique con un dispositivo externo, como un Arduino u otro controlador electrónico.
- Liderar una dinámica para asignar roles SCRUM entre los equipos de tal modo que un equipo sea el cliente de otro equipo.
- Solicitar una tabla comparativa de aplicaciones populares en el mercado regional.
- Generar una dinámica de licitación de proyectos para seleccionar el proyecto que desarrollará cada equipo.
- Efectuar el cierre de ciclos de aprendizaje no solamente al concluir cada tema o subtema, sino de cada sesión de clase, con la finalidad de lograr un proceso lógico de enseñanza-aprendizaje, en el que el alumno pueda apreciar tanto sus logros cotidianos y la importancia de su esfuerzo y constancia, como la importancia de la afirmación de sus capacidades para dar paso a la adquisición de nuevas competencias, especialmente las relacionadas con el manejo de tecnologías de información y la comunicación para procesar u obtener datos, así como expresar ideas.
- Organizar sistemáticamente la información que se ha de manejar y procesar para su aprendizaje. Efectuando explícitamente la vinculación de esta unidad con la que precede, con el fin de que el alumno valore su importancia académica y curricular.
- Promover la dinámica grupal colaborativa y cooperativa a través de la realización de las técnicas didácticas y de aprendizaje correspondientes, durante el transcurso de cada sesión para favorecer el clima que fomente el intercambio constructivo de ideas.

Estrategias de Aprendizaje	Recursos Académicos
<ul style="list-style-type: none"> • Realizar de manera individual investigación sobre la carpeta de recursos en una aplicación Android. • Desarrollar, con un compañero de dupla, una aplicación que despliegue contenido multimedia: imágenes, audio o video. • Realizar con el profesor actividades en el laboratorio de cómputo para desarrollar animaciones de imágenes. • Desarrollar la práctica núm. 9 “Presentación de multimedia con drag and drop”. • Desarrollar la práctica núm. 10 “Animación”. • Desarrollar la práctica núm. 11. “Interacción con mapas”. • Desarrollar la práctica núm. 12 “Bases de Datos”. • Realizar la actividad de evaluación 3.1.1 • Realizar un cuadro semántico con la estructura de documentos XML y json. • Investigar sobre las clases Document y DocumentParser para la interpretación de documentos XML y json. • Desarrollar práctica núm. 13 “Lector de RSS”. • Presentar ante clase una infografía sobre el tema Internet de las Cosas. • Desarrollar la práctica núm 14 “Interacción con Arduino”. • Realizar la actividad de evaluación 3.2.1 • Participar en las dinámicas de grupo para generar los equipos SCRUM que desarrollarán el proyecto integrador. • Realizar la actividad de evaluación 3.3.1 	<p>Software Software Eclipse Mars o superior, o Android Studio 2.0 o superior</p> <ul style="list-style-type: none"> • https://developer.android.com/guide/topics/media/camera.html • http://andruino2k14.blogspot.mx/2014/04/tutoria-l-carro-wi-fi-4x4-con-android.html • http://robologs.net/2015/10/29/tutorial-de-arduino-bluetooth-y-android-2-crear-una-app-con-mit-inventor/ • https://developer.android.com/guide/topics/graphics/index.html • https://developer.android.com/guide/topics/data/data-storage.html#db <p>Desarrollo de un app con Arduino. Recuperado el 15/12/17 de https://www.youtube.com/watch?v=S8q1vbVMV2Q</p> <p>Lector de RSS con RecyclerView. Recuperado el 15/12/17 de https://www.youtube.com/watch?v=19NSIa4FNdQ&index=1&list=PLOvzGCa-rsH-9QjIFBVHfBNUzPGHGzj-5</p>

6. Prácticas/Actividades

Unidad de aprendizaje:	Administración del ciclo de desarrollo de aplicaciones	Número:	1
Práctica:	Diseño de Primera aplicación: “¡Hola Mundo!”	Número:	1
Propósito de la práctica:	Conocer el ambiente de desarrollo Android y los componentes de una app		
Escenario:	Laboratorio de Cómputo	Duración	3 horas

Materiales, herramientas, instrumental, maquinaria y equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo Android Studio o Eclipse Simulador de dispositivo móvil o Dispositivo Android y cable USB 	<p>Desarrolla utilizando Android Studio una aplicación en la cual aparezca la frase “¡Hola mundo!” cuando presiones un botón.</p> <p>Instrucciones:</p> <ul style="list-style-type: none"> Abrir Android Studio y crear un proyecto llamado ‘HolaMundo’. Crear una nueva actividad en blanco. Agregar un botón en la pantalla y cambiar el texto de este para que diga “Presióname”. Una vez que se ha agregado el botón, asegurarse que se encuentre centrado en la pantalla, aun cuando ésta rote. (Tip: Se puede utilizar la propiedad “Gravity”. Agregar un “TextView” encima del botón. Por el momento este textview debe estar vacío. En MainActivity.java, declarar los elementos que se van a utilizar utilizando ‘private Button’ y ‘private ‘TextView’. Crear una función que permita cambiar el texto a “¡Hola mundo!” cuando se presione el botón. Para esto es necesario utilizar un “Event Listener”, el cual va a ser el encargado de percatar cuando el botón sea presionado. <p> ADVERTENCIA DE RIESGO ELECTRICO.</p>

Unidad de Aprendizaje:	Administración del ciclo de desarrollo de aplicaciones	Número:	1
Práctica:	Manejo de evento de calculadora numérica	Número:	2
Propósito de la práctica:	Comprender el manejo de eventos y el uso de Layouts en Apps Android		
Escenario:	Laboratorio de Cómputo	Duración	4 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo • Android Studio o Eclipse • Simulador de dispositivo móvil o Dispositivo Android y cable USB 	<p>Desarrollo de una calculadora que pueda hacer operaciones básicas como suma, resta, multiplicación y división, utilizando Android Studio.</p> <p>Instrucciones:</p> <ul style="list-style-type: none"> • Abrir Android Studio y crear un nuevo proyecto llamado “Calculadora”. • Crear una nueva actividad en blanco. • En app > res > layout > activity_main.xml, en la vista de diseño, agregar un “Large Text” que diga “Mi Calculadora”. Centrar el texto en la parte superior de la pantalla. • Agregar un “Medium Text” que diga “Número 1”. Junto a éste agregar un “Text Field” de Número y cambiar su id a “etPrimerNumero”. • Repetir el paso anterior una vez más para agregar un segundo número. En estos cuadros de texto se insertarán los valores para hacer las operaciones. • Una vez que se tienen los elementos anteriores, se tienen que agregar cuatro botones (uno para cada una de las operaciones básicas que realizará la calculadora). Cambiar el texto que se muestra por cada uno de los símbolos (+, -, *, /) y sus IDs, esto último para que sea más sencillo poder identificarlos al realizar el código. • Por último en la vista, agregar un “Large Text” que diga “Resultado:” • Para la parte de la programación, en MainActivity.java se deben declarar todos los elementos que se utilizan en la vista previa utilizando: ‘private Button’, ‘private TextView’ y ‘private EditText’. • Después se debe crear un método para inicializar cada uno de los componentes de la calculadora. Ejemplo: btnSuma=(Button)findViewById(R.id.btnSuma); • Dentro de éste método se deben agregar los “Event Listeners”, los cuales van a ser los encargados de percatar cuando se presione algún botón. Nota: No olvidar poner

en la clase: `public class MainActivity extends ActionBarActivity implements`

View.OnClickListener

Ejemplo: `btnSuma.setOnClickListener(this)`.

- Por último, para que la acción se pueda llevar a cabo, se debe crear una función “`onClick(View view)`”, la cual va a realizar las operaciones correspondientes dependiendo del botón que se presione. Dentro de éste método se tienen que hacer evaluaciones para saber qué operación se va a realizar. (Tip: Se puede utilizar un *switch*).

Nota: Tomar en cuenta los diferentes casos que puede haber en una división.



ADVERTENCIA DE RIESGO ELECTRICO.

Nombre del Alumno:	
Unidad de Aprendizaje:	1 Administración del ciclo de desarrollo de aplicaciones
Resultado de Aprendizaje:	1.2 Utiliza estándares de administración de proyectos utilizando el modelo de programación ágil para la optimización del ciclo de vida de desarrollo de software con el fin de aumentar la calidad del proceso y del producto en el desarrollo de aplicaciones
Actividad núm. 3:	Simulación de jornada SCRUM


Para esta actividad, seguirás las indicaciones de tu profesor.

- Integra con tu equipo los diversos roles:
 - Cliente
 - SCRUM master
 - Desarrolladores
- El profesor definirá una actividad a realizar



- El cliente definirá las especificaciones mínimas del proyecto
- Cliente y equipo definirán las historias de usuario
- El SCRUM master definirá las actividades del *backlog*
- Con indicaciones del profesor, se correrán tres *sprints*, ajustando las actividades según los resultados

Unidad de Aprendizaje:	Administración del ciclo de desarrollo de aplicaciones	Número:	1
Práctica:	Desarrollo con SCRUM	Número:	4
Propósito de la práctica:	Conocer el proceso de desarrollo bajo la metodología SCRUM		
Escenario:	Laboratorio de Cómputo	Duración	4 horas

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo • Android Studio o Eclipse • Simulador de dispositivo móvil o Dispositivo Android y cable USB • Tarjetas de Papel • Pizarrón • Notas adhesivas 	<p>Para esta actividad, sigue las indicaciones del profesor.</p> <ol style="list-style-type: none"> 1. Cada equipo definirá un pequeño proyecto a realizar, y fungirá como cliente de otro equipo 2. Define los roles de SCRUM Master y asociados. 3. Define las historias de usuario y las actividades del <i>backlog</i> 4. Realiza al menos tres <i>sprints</i> de desarrollo. 5. Valida los resultados con el equipo cliente. 6. Evalúa la calidad del proceso según los resultados logrados. <p>Es importante notar que en esta actividad se busca la comprensión del modelo SCRUM, no es imperativo terminar el app en el tiempo asignado para la actividad</p> <p> ADVERTENCIA DE RIESGO ELECTRICO.</p>

Unidad de Aprendizaje:	Programación de la interfaz de la aplicación	Número:	2
-------------------------------	--	----------------	---

Práctica:	Creación ventanas de actividades interconectadas	Número:	5
------------------	--	----------------	---

Propósito de la práctica:	Crear ventanas de actividades interconectadas en una aplicación
----------------------------------	---

Escenario:	Laboratorio de Cómputo	Duración	4 horas
-------------------	------------------------	-----------------	---------

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo • Android Studio o Eclipse • Simulador de dispositivo móvil o Dispositivo Android y cable USB 	<p>Crear una aplicación que por medio de varias “Text Box” y la clase ‘Intent’, permita que el usuario pueda dar información para que ésta pueda ser luego usada en otra pantalla.</p> <ul style="list-style-type: none"> • Abrir Android Studio y crear un nuevo proyecto con el nombre “JuegoPalabras”. • Crear una nueva actividad en blanco. • En seguida crear una segunda actividad en blanco, la cual será la segunda pantalla. • Dentro de la segunda actividad, crear una pequeña historia utilizando diferentes “Text Views”. En esta historia, de deben dejar espacios en blanco, los cuales van a ser rellenados por el usuario. • En la primera actividad se deben insertar “textViews” por cada una de las palabras que se necesitan escribir en la historia. En cada uno de estos “text Views” hay que escribir el tipo de palabra que es, ya sea sustantivo, pronombre, verbo en alguna conjugación, adverbio, etc. • Nota: Es muy importante mantener el orden correspondiente de cada una de las palabras. • Junto a cada tipo de palabra se debe agregar un “textBox” y cambiar su id para poder tener un mejor control sobre ellos. • Debajo de los “textViews” y los “textBox”, agregar un botón que diga “Crear historia”. • En el código de MainActivity.java, dentro del método ‘protected void onCreate (Bundle savedInstanceState)’ escribir: final EditText editText = (EditText)findViewById(R.id._____) En el espacio en blanco, se debe de poner el id de cada uno de los elementos.

- Dentro del mismo método se debe declarar la variable del botón e instanciar un “Event Listener” del mismo.
- Una vez que se tiene el ‘Event Listener’, lo que se tiene que hacer es crear un método en el cual se declare un objeto de la clase Intent, el cual permitirá recolectar los datos que se tengan en la primera pantalla para luego pasarla a la segunda.
- Ejemplo: `Intent intent = new Intent(getApplicationContext(), SegundaActividad.class);`
 - `intent.putExtra(“sust”, sustantivo);`
 - `startActivity(intent);`
- Cuando se termine de realizar el método en MainActivity.java, en SegundaActividad.java se tienen que declarar cada una de los elementos que se utilizan, y un objeto de la clase Intent.
- Crear un método que permita colocar cada uno de los datos extraídos en la primera pantalla, utilizando el método `getIntent()`, para después colocarlos en la variable correcta.

Ejemplo: `String sustantivo = intent.getStringExtra(“sust”);`



ADVERTENCIA DE RIESGO ELECTRICO.

Unidad de Aprendizaje:	Programación de la interfaz de la aplicación	Número:	2
-------------------------------	--	----------------	---

Práctica:	Modificación de imágenes como respuesta a evento	Número:	6
------------------	--	----------------	---

Propósito de la práctica:	Comprender el uso de funciones para el manejo de eventos generados por el usuario
----------------------------------	---

Escenario:	Laboratorio de Cómputo	Duración	4 horas
-------------------	------------------------	-----------------	---------

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo • Android Studio o Eclipse • Simulador de dispositivo móvil o Dispositivo Android y cable USB 	<p>Crear una aplicación en la cual las imágenes vayan cambiando cada vez que el usuario presione un botón.</p> <ul style="list-style-type: none"> • Abrir Android Studio y crear un nuevo proyecto llamado “Cambiolimagen”. • Crear una actividad en blanco. • En app > res > layout > activity_main.xml, en la vista de diseño, agregar un “ImageSwitcher” en la vista de pantalla. En las propiedades del elemento, cambiar las dimensiones en para que no excedan las dimensiones de la pantalla, y también cambiar su id. • Una vez hecho esto, agregar un botón que diga “Presióname” debajo de la imagen y cambiar su id. • Después se deben copiar las imágenes que se van a utilizar y pegarlas en la carpeta res > drawable. • Una vez que se ha terminado de diseñar la vista, en MainActivity.java se debe declarar en un arreglo las imágenes que se van a utilizar. Ejemplo: <code>int numImagen[] = [R.drawable.img1, R.drawable.img2, ...];</code> • Para poder manejar los elementos que se encuentran en el arreglo se deben crear dos contadores, uno que permita saber la posición de la imagen en el arreglo y otro para saber el número de imágenes. • Crear un método para agregar las imágenes dinámicamente, y crear un “Event Listener” que permita hacer el cambio de imagen cuando se presione el botón. <code>public void onClick(View view).</code>

Nota: Dentro de este método se va a ir contando en qué posición del arreglo se encuentra.

Nota 2: Se tienen que repetir las imágenes, es decir, cuando se llegue a la última imagen, al presionar el botón tiene que mostrar de nuevo la primera.



ADVERTENCIA DE RIESGO ELECTRICO.

Unidad de Aprendizaje:	Programación de la interfaz de la aplicación	Número:	2
-------------------------------	--	----------------	---

Práctica:	Manejo de acelerómetro del dispositivo	Número:	7
------------------	--	----------------	---

Propósito de la práctica:	Utilizar las funciones de Android para lectura de sensores del dispositivo
----------------------------------	--

Escenario:	Laboratorio de Cómputo	Duración	4 horas
-------------------	------------------------	-----------------	---------

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo • Android Studio o Eclipse • Simulador de dispositivo móvil o Dispositivo Android y cable USB 	<p>Crear una aplicación en la cual se modifiquen los datos dependiendo de los movimientos que haga el teléfono.</p> <ul style="list-style-type: none"> • Crear un proyecto nuevo con el nombre “Acelerometro” en Android Studio. • Crear una nueva actividad en blanco • Dentro de esta actividad en app > res > layout > activity_main.xml, en la vista de diseño, agregar tres “TextViews”, en los cuales estarán contenidos los valores X,Y y Z. Definir el id de cada uno de los elementos. • Una vez que se tienen estos elementos en la pantalla, ir a MainActivity.java e importar las siguientes funciones para poder utilizar los sensores del teléfono: <code>import android.hardware.Sensor</code> <code>import android.hardware.SensorManager</code> <code>import android.hardware.SensorEvent</code> <code>import android.hardware.SensorEventListener</code> <code>import android.widget.TextView</code> • Después se tienen que declarar los elementos que se van a utilizar. En estos elementos también se tienen que incluir los objetos de clase Sensor y SensorManager. Nota: Recuerda que se tiene que implementar la clase <code>SensorEventListener</code> • Instanciar el objeto SensorManager: <code>SM = (SensorManager) getSystemService(SENSOR_SERVICE);</code>

- Crear el sensor del acelerómetro: `miSensor = SM.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);`
- Crear el "Sensor Listener" del sensor: `SM.registerListener(this, miSensor, SensorManager.SENSOR_DELAY_NORMAL);`
- Asignar los valores de los "TextViews" a las variables.
- Una vez que se tienen los valores de las variables, es necesario implementar los métodos de la clase `SensorEventListener`: `onAccuracyChanged(Sensor sensor, in tac)` y `onSensorChanged(SensorEvent event)`.
Nota: el primero no se va a utilizar, sin embargo se tiene que instanciar debido a que es un método de la clase `SensorEventListener`.
- Modificar el método `onSensorChanged()` para que cada vez que se modifiquen los datos en X,Y y Z, éstos se vean reflejados en la pantalla principal.



ADVERTENCIA DE RIESGO ELECTRICO.

Unidad de Aprendizaje:	Programación de la interfaz de la aplicación	Número:	2
-------------------------------	--	----------------	---

Práctica:	Lector de RSS parte 1. Fragmentos	Número:	8
------------------	-----------------------------------	----------------	---

Propósito de la práctica:	Conocer los elementos de Android para manejo de datos repetitivos		
----------------------------------	---	--	--

Escenario:	Laboratorio de Cómputo	Duración	4 horas
-------------------	------------------------	-----------------	---------

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo • Android Studio o Eclipse • Simulador de dispositivo móvil o Dispositivo Android y cable USB 	<p>Crear una aplicación con un lector RSS que permita ver las clases que se van a impartir durante el ciclo escolar y los profesores de cada una de ellas.</p> <p>En esta primera parte crearemos los layouts para contener los datos que se leerán del RSS</p> <ul style="list-style-type: none"> • Abrir Android Studio y crear un nuevo proyecto con el nombre “LectorRSS”. • Crear una nueva actividad en blanco. • El siguiente paso es diseñar la pantalla principal en donde se va a mostrar la lista de los elementos, en este caso los elementos del feed. Utiliza un ListView o RecyclerView • Después, se tiene que crear la segunda pantalla, en donde se mostrarán los detalles de la clase: El profesor que la impartirá, el horario y el salón. Esto debe ser un ListItem o CardView • Una vez que se tienen las dos pantallas, se debe crear una clase “Modelo de Clase”, la cual se utilizará para poder guardar las clases en un <i>ArrayList</i>. • Dentro de MainActivity.java, se declaran los campos que se van a utilizar en la pantalla principal, donde se va a mostrar los nombres de las clases. • Dentro de un nuevo método llamado <i>public void onCreate(Bundle savedInstanceState)</i> inicializar los campos. • Una vez que ya se tienen todos los elementos, se debe crear un método <i>leerDatos()</i>, el cual será el método encargado de descargar los datos de internet y después almacenarlos. Para esta primera parte se crearán datos de manera estática.

- En SegundaActividad.java, se deben declarar las variables o los elementos que se van a utilizar en la clase de la segunda pantalla en los cuales se van a mostrar los datos.
Ejemplo: `private TextView txtClase;`
`private TextView txtProfesor;`
- Estos elementos se tienen que inicializar en el método `onCreate()` y dentro de éste se tienen que realizar las instrucciones necesarias para mostrar los datos.
- Por último, se tiene que crear un método, el cual permita obtener los datos de las clases para poder mostrárselos al usuario.



ADVERTENCIA DE RIESGO ELECTRICO.

Unidad de Aprendizaje:	Programación de interacciones con datos externos	Número:	3
-------------------------------	--	----------------	---

Práctica:	Presentación de recursos multimedia con evento Drag and drop	Número:	9
------------------	--	----------------	---

Propósito de la práctica:	Acceder a la carpeta de recursos de una aplicación
----------------------------------	--

Escenario:	Laboratorio de Cómputo	Duración	4 horas
-------------------	------------------------	-----------------	---------

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo • Android Studio o Eclipse • Simulador de dispositivo móvil o Dispositivo Android y cable USB 	<p>Esta práctica permite conocer el manejo de recursos locales.</p> <ul style="list-style-type: none"> • Abrir Android Studio y crear un proyecto nuevo con el nombre “DragAndDrop”. • Crear una nueva actividad en blanco. • Agregar un “ImageView” a la pantalla principal. • Agregar una imagen a la carpeta de recursos, y nombrarla de manera apropiada para poder accederla desde la clase Java. • Modificar el siguiente código de MainActivity.java para que la aplicación logre hacer que el usuario pueda mover la “ImageView” a cualquier lugar de la pantalla, y agregar los elementos necesarios a la pantalla principal: <pre>import android.app.Activity; import android.os.Bundle; import android.view.MotionEvent; import android.view.View; import android.view.ViewGroup; import android.view.View.OnClickListener; import android.widget.ImageView; import android.widget.RelativeLayout; public class MainActivity extends Activity { private ImageView private ViewGroup</pre>

```
private int xDelta;
private int yDelta;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout._____);
    rootLayout = (ViewGroup)findViewById(R.id._____);
    img = (ImageView)rootlayout.findViewById(R.id._____);
    RelativeLayout.LayoutParams layoutParams = new
RelativeLayout.LayoutParams(_____, _____);
img.setLayoutParams(_____);
img.setOnTouchListener(new _____);
}

private final class ChoiceTouchListener implements OnTouchListener {
    public Boolean onTouch(View view, MotionEvent event){
        final int X = (int) event._____;
        final int Y = (int) event._____;
        switch (event.getAction() & MotionEvent.ACTION_MASK) {
            case MotionEvent.ACTION_DOWN:
                RelativeLayout.LayoutParams lParams =
(RelativeLayout.LayoutParams)_____._____;
                xDelta = _____;
                yDelta = _____;
                break;
            case MotionEvent._____;
                break;
            case MotionEvent._____;
                break;
            case MotionEvent._____;
                break;
            case MotionEvent.ACTION_MOVE:
```

```
RelativeLayout.LayoutParams layoutParams =  
(RelativeLayout.LayoutParams) _____;  
layoutParams.leftMargin = _____;  
layoutParams.topMargin = _____;  
layoutParams.rightMargin = _____;  
layoutParams.bottomMargin = _____;  
view.setLayoutParams(layoutParams);  
break;  
}  
rootLayout.invalidate( );  
return true;  
}  
}
```




ADVERTENCIA DE RIESGO ELECTRICO.

Unidad de Aprendizaje:	Programación de interacciones con datos externos	Número:	3
-------------------------------	--	----------------	---

Práctica:	Animación de elementos gráficos	Número:	10
------------------	---------------------------------	----------------	----

Propósito de la práctica:	Conocer los comandos Java para animación de objetos en una aplicación móvil		
----------------------------------	---	--	--

Escenario:	Laboratorio de Cómputo	Duración	4 horas
-------------------	------------------------	-----------------	---------

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo • Android Studio o Eclipse • Simulador de dispositivo móvil o Dispositivo Android y cable USB 	<p>Genera una aplicación que presente una imagen y la anime con su evento onclick.</p> <ol style="list-style-type: none"> 1. Debe haber al menos una imagen en la pantalla 2. Utiliza propiedades ViewAnimation o DrawableAnimation para animar tu objeto gráfico 3. Modifica la clase Java para utilizar las propiedades anteriores 4. Agrega un botón y asigna una función que ejecute el código de ViewAnimation o DrawableAnimation para lograr la animación al orpimir el botón <p> ADVERTENCIA DE RIESGO ELECTRICO.</p>

Unidad de Aprendizaje:	Programación de interacciones con datos externos	Número:	3
-------------------------------	--	----------------	---

Práctica:	Interacción con mapas	Número:	11
------------------	-----------------------	----------------	----

Propósito de la práctica:	Utilizar comandos Java para interactuar con APIs externos
----------------------------------	---

Escenario:	Laboratorio de Cómputo	Duración	4 horas
-------------------	------------------------	-----------------	---------

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> Equipo de cómputo Android Studio o Eclipse Simulador de dispositivo móvil o Dispositivo Android y cable USB 	<p>Crear una aplicación que permite colocar un marcador en una aplicación creada utilizando Google Maps.</p> <ul style="list-style-type: none"> El primer paso que se debe realizar es abrir Android Studio e iniciar SDK Manager para instalar los servicios de Google Play. Una vez que se ha hecho esto, se debe crear un proyecto nuevo con el nombre “Marcador”. Crear una nueva actividad “Actividad de Google Maps” Entrar a <i>Google Developer Console</i> en un navegador utilizando el URL: <code>console.developers.google.com</code> y crear un nuevo proyecto. Después abrir la Opción “APIs & auth” se debe hacer clic en <i>Credentials</i> y crear una API Key para Android. Una vez que se tiene esto, se debe copiar la huella certificada de Android Studio en el <i>Google Developer Console</i>. El siguiente paso es crear y copiar la API Key que se obtuvo en Android Studio y activar el API para el proyecto que se va a realizar. <p>Ejemplo:</p> <pre><string name= “google_maps_key” translatable= “false” templateMergeStrategy=”preserve”> API KEY </string></pre>

- En Android Studio, ir a AndroidManifest.xml y activar los permisos necesarios de Google Play (Internet, posición, etc.)
Ejemplo: `<uses-permission android:name = "android.permission.INTERNET" />`
- Por último en MapsActivity.java en el método *private void setUpMap()* se debe añadir el título de las coordenadas de Google Maps del lugar en el cual se quiere poner el marcador.



ADVERTENCIA DE RIESGO ELECTRICO.

Unidad de Aprendizaje:	Programación de interacciones con datos externos	Número:	3
-------------------------------	--	----------------	---

Práctica:	Manejo de bases de datos internas	Número:	12
------------------	-----------------------------------	----------------	----

Propósito de la práctica:	Utilizar clases de Java para acceder a bases de datos privadas en la aplicación		
----------------------------------	---	--	--

Escenario:	Laboratorio de Cómputo	Duración	4 horas
-------------------	------------------------	-----------------	---------

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo • Android Studio o Eclipse • Simulador de dispositivo móvil o Dispositivo Android y cable USB 	<p>El objetivo es crear una aplicación que pueda generar y consumir datos en una base SQLite interna.</p> <ul style="list-style-type: none"> • Abrir Android Studio y crear un nuevo proyecto llamado "BasesDeDatos" • Crear una nueva actividad en blanco. • Crear una clase nueva para poder manejar SQLite, que es la base de datos. Para hacer esto debes hacer clic derecho en el paquete en donde se encuentra MainActivity.java > nuevo > Java Class y ponerle como nombre DatabaseHelperClass. • En esta clase debe ser una extensión de SQLiteOpenHelper. <i>Public class DatabaseHelper extends SQLiteOpenHelper.</i> Una vez realizado este paso, se deben implementar todos los constructores y los métodos de SQLiteOpenHelper. Nota: Para que SQLiteOpenHelper funcione se tiene que escribir al inicio: <i>import android.database.sqlite.SQLiteOpenHelper</i> • Para crear la base de datos se deben declarar las variables en donde se encontrará el nombre de la base de datos(DATABASE_NAME), el nombre de la(s) tabla(s) (TABLE_NAME) y el nombre de las columnas de la tabla (COL_#). Nota. El nombre de la base de datos siempre debe terminar en .db • Crear la base de datos en el constructor: • <pre>public DatabaseHelper(Context context){ super(context, DATABASE_NAME, null, 1); }</pre>

- Sobrecribir el método para agregar un registro a la base de datos
- Sobrecribir el método para recuperar todos los registros de la base de datos.
- Generar una ventana de actividades que utilice el método de creación de registro
- Generar una ventana de actividades que utilice el método de recuperación de registros. Esta ventana puede mostrar el primer registro disponible o utilizar un ListView para mostrar todos los registros recuperados




ADVERTENCIA DE RIESGO ELECTRICO.

Unidad de Aprendizaje:	Programación de interacciones con datos externos	Número:	3
-------------------------------	--	----------------	---

Práctica:	Lector de RSS parte 2	Número:	13
------------------	-----------------------	----------------	----

Propósito de la práctica:	Utilizar las clases de Java para leer flujos de datos desde servicios Web
----------------------------------	---

Escenario:	Laboratorio de Cómputo	Duración	4 horas
-------------------	------------------------	-----------------	---------


Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo • Android Studio o Eclipse • Simulador de dispositivo móvil o Dispositivo Android y cable USB • Navegador web 	<p>Esta práctica concluye la aplicación de un lector de RS</p> <ol style="list-style-type: none"> 1. Recupera tu proyecto de Lector de RSS 2. Genera una clase asíncrona que contendrá el código del lector de RSS 3. Utiliza la clase XMLReader para abrir un feed RSS, por ejemplo, puedes leer este feed: http://science.sciencemag.org/rss/current.xml 4. Utiliza la clase Document para interpretar los contenidos del feed RSS 5. Modifica el Adaptador de la práctica pasada para que rellene los fragmentos utilizando los datos del RSS <p> ADVERTENCIA DE RIESGO ELECTRICO.</p>

Unidad de Aprendizaje:	Programación de interacciones con datos externos	Número:	3
-------------------------------	--	----------------	---

Práctica:	Interacción con tarjeta Arduino	Número:	14
------------------	---------------------------------	----------------	----

Propósito de la práctica:	Realizar conexiones entre el dispositivo móvil y dispositivos externos		
----------------------------------	--	--	--

Escenario:	Laboratorio de Cómputo	Duración	4 horas
-------------------	------------------------	-----------------	---------

Materiales, Herramientas, Instrumental, Maquinaria y Equipo	Desempeños
<ul style="list-style-type: none"> • Equipo de cómputo • Android Studio o Eclipse • Dispositivo Android con capacidad Bluetooth y cable USB • Tarjeta Arduino • Shield Bluetooth para Arduino 	<p>Realiza una aplicación que interactúe con un kit Arduino.</p> <ol style="list-style-type: none"> 1. Genera un proyecto nuevo 2. Importa las librerías para tener acceso a Bluetooth 3. Genera un código para detectar un evento click en la pantalla de tu aplicación 4. Genera el código del Listener que envía información por el puerto Bluetooth 5. Genera el código de Arduino para detectar la entrada en Bluetooth <p>Puedes utilizar como guía, por ejemplo, la información en http://www.instructables.com/id/Android-Bluetooth-Control-LED-Part-2/?ALLSTEPS</p> <p> ADVERTENCIA DE RIESGO ELECTRICO.</p>

II. Guía de evaluación del módulo Diseño y desarrollo de aplicaciones móviles

7. Descripción

La guía de evaluación es un documento que define el proceso de recolección y valoración de las evidencias requeridas por el módulo desarrollado y tiene el propósito de guiar en la evaluación de las competencias adquiridas por los alumnos, asociadas a los Resultados de Aprendizaje; en donde, además, describe las técnicas y los instrumentos a utilizar y la ponderación de cada actividad de evaluación. Los Resultados de Aprendizaje se definen tomando como referentes: las **competencias genéricas** que va adquiriendo el alumno para desempeñarse en los ámbitos personal y profesional que le permitan convivir de manera armónica con el medio ambiente y la sociedad; las **disciplinares**, esenciales para que los alumnos puedan desempeñarse eficazmente en diversos ámbitos, desarrolladas en torno a áreas del conocimiento y las **profesionales** que le permitan un desempeño eficiente, autónomo, flexible y responsable de su ejercicio profesional y de actividades laborales específicas, en un entorno cambiante que exige la multifuncionalidad.

Durante el proceso de enseñanza - aprendizaje es importante considerar tres finalidades de evaluación: **diagnóstica, formativa y sumativa**.

La evaluación **diagnóstica** nos permite establecer un **punto de partida** fundamentado en la detección de la situación en la que se encuentran nuestros alumnos. Permite también establecer vínculos socio-afectivos entre el docente y su grupo. El alumno a su vez podrá obtener información sobre los aspectos donde deberá hacer énfasis en su dedicación. El docente podrá **identificar las características del grupo y orientar adecuadamente sus estrategias**. En esta etapa pueden utilizarse mecanismos informales de recopilación de información.

La evaluación **formativa** se realiza durante todo el proceso de aprendizaje del alumno, en forma constante, ya sea al finalizar cada actividad de aprendizaje o en la integración de varias de éstas. Tiene como finalidad **informar a los alumnos de sus avances** con respecto a los aprendizajes que deben alcanzar y advertirle sobre dónde y en qué aspectos tiene debilidades o dificultades para poder regular sus procesos. Aquí se admiten errores, se identifican y se corrigen; es factible trabajar colaborativamente. Asimismo, el docente puede asumir nuevas estrategias que contribuyan a mejorar los resultados del grupo.

Finalmente, la evaluación **sumativa** es adoptada básicamente por una función social, ya que mediante ella se asume una acreditación, una promoción, un fracaso escolar, índices de deserción, etc., a través de **criterios estandarizados y bien definidos**. Las evidencias se elaboran en forma individual, puesto que se está asignando, convencionalmente, un criterio o valor. Manifiesta la síntesis de los logros obtenidos por ciclo o período escolar.

Con respecto al agente o responsable de llevar a cabo la evaluación, se distinguen tres categorías: la **autoevaluación** que se refiere a la valoración que hace el alumno sobre su propia actuación, lo que le permite reconocer sus posibilidades, limitaciones y cambios necesarios para mejorar su aprendizaje. Los roles de evaluador y evaluado coinciden en las mismas personas

La **coevaluación** en la que los alumnos se evalúan mutuamente, es decir, evaluadores y evaluados intercambian su papel alternativamente; los alumnos en conjunto, participan en la valoración de los aprendizajes logrados, ya sea por algunos de sus miembros o del grupo en su conjunto; La coevaluación permite al alumno y al docente:

- Identificar los logros personales y grupales
- Fomentar la participación, reflexión y crítica constructiva ante situaciones de aprendizaje
- Opinar sobre su actuación dentro del grupo
- Desarrollar actitudes que se orienten hacia la integración del grupo
- Mejorar su responsabilidad e identificación con el trabajo
- Emitir juicios valorativos acerca de otros en un ambiente de libertad, compromiso y responsabilidad

La **heteroevaluación** que es el tipo de evaluación que con mayor frecuencia se utiliza, donde el docente es quien, evalúa, su variante externa, se da cuando agentes no integrantes del proceso enseñanza-aprendizaje son los evaluadores, otorgando cierta objetividad por su no implicación.

Cada uno de los Resultados de Aprendizaje (RA) tiene asignada al menos una actividad de evaluación (AE), a la cual se le ha determinado una ponderación con respecto a la Unidad a la cual pertenece. Ésta a su vez, tiene una ponderación que, sumada con el resto de Unidades, **conforma el 100%**. Es decir, para considerar que se ha adquirido la competencia correspondiente al módulo de que se trate, deberá **ir acumulando** dichos porcentajes a lo largo del período para estar en condiciones de acreditar el mismo. Cada una de estas ponderaciones dependerá de la relevancia que tenga dicha actividad con respecto al RA y éste a su vez, con respecto a la Unidad de Aprendizaje.

La ponderación que se asigna en cada una de las actividades queda asimismo establecida en la **Tabla de ponderación**, la cual está desarrollada en una hoja de cálculo que permite, tanto al alumno como al docente, ir observando y calculando los avances en términos de porcentaje, que se van alcanzando. Esta tabla de ponderación contiene los Resultados de Aprendizaje y las Unidades a las cuales pertenecen. Asimismo, indica, en la columna de actividades de evaluación, la codificación asignada a ésta desde el programa de estudios y que a su vez queda vinculada al Sistema de Evaluación Escolar SAE. Las columnas de aspectos a evaluar, corresponden al tipo de aprendizaje que se evalúa: **C = conceptual; P = Procedimental y A = Actitudinal**. Las siguientes tres columnas indican, en términos de porcentaje: la primera el **peso específico** asignado desde el programa de estudios para esa actividad; la segunda, **peso logrado**, es el nivel que el alumno alcanzó con base en las evidencias o desempeños demostrados; la tercera, **peso acumulado**, se refiere a la suma de los porcentajes alcanzados en las diversas actividades de evaluación y que deberá acumular a lo largo del ciclo escolar.

Otro elemento que complementa a la matriz de ponderación es la **rúbrica o matriz de valoración**, que establece los **indicadores y criterios** a considerar para evaluar, ya sea un producto, un desempeño o una actitud. Una matriz de valoración o rúbrica es, como su nombre lo indica, una matriz de doble entrada en la cual se establecen, por un lado, los **indicadores** o aspectos específicos que se deben tomar en cuenta como **mínimo indispensable** para evaluar si se ha logrado el resultado de aprendizaje esperado y, por otro, los criterios o **niveles de calidad o satisfacción alcanzados**. En las celdas centrales se describen los criterios que se van a utilizar para evaluar esos indicadores, explicando cuáles son las características de cada uno. Los criterios que se han establecido son: **Excelente**, en el cual, además de cumplir con los estándares o requisitos establecidos como necesarios en el logro del producto o desempeño, es propositivo, demuestra iniciativa y creatividad, o que va más allá de lo que se le solicita como mínimo, aportando elementos adicionales en pro del indicador; **Suficiente**, si cumple con los estándares o requisitos establecidos como necesarios para demostrar que se ha desempeñado adecuadamente en la actividad o elaboración del producto. Es en este nivel en el que podemos decir que se ha adquirido la competencia. **Insuficiente**, para cuando no cumple con los estándares o requisitos mínimos establecidos para el desempeño o producto.

8. Tabla de ponderación

UNIDAD	RA	ACTIVIDAD DE EVALUACIÓN	ASPECTOS A EVALUAR			% Peso Específico	% Peso Logrado	% Peso Acumulado
			C	P	A			
1. Administración del ciclo de desarrollo de aplicaciones	1.1 Identifica ambientes de desarrollo empleando herramientas Android dentro del lenguaje de programación Java para el desarrollo eficiente de aplicaciones móviles	1.1.1	▲	▲	▲	15%		
	1.2 Utiliza estándares de administración de proyectos utilizando el modelo de programación ágil para la optimización del ciclo de vida de desarrollo de software con el fin de aumentar la calidad del proceso y del producto en el desarrollo de aplicaciones	1.2.1	▲	▲	▲	15%		
% PESO PARA LA UNIDAD						30%		
2. Programación de la interfaz de la aplicación	2.1 Utiliza las estructuras Activity e Intent del lenguaje Java para la creación de la estructura de navegación de una aplicación móvil para el desarrollo eficiente de aplicaciones móviles	2.1.1	▲	▲	▲	20%		
	2.2 Integra estructuras de Layout y Fragmentos en Java para crear interfaces fluidas empleando sus herramientas que permitan el despliegue de datos de forma variable en una aplicación	2.2.1	▲	▲	▲	20%		
% PESO PARA LA UNIDAD						40%		

3. Programación de interacciones con datos externos	3.1 Utiliza elementos del lenguaje Java empleando sus herramientas para crear aplicaciones que interactúen con elementos multimedia y bases de datos locales.	3.1.1	▲	▲	▲	10%		
	3.2 Utiliza componentes del lenguaje Java conforme sus procedimientos para la creación de aplicaciones que consuman datos externos por medio de servicios web.	3.2.1	▲	▲	▲	10%		
	3.3 Desarrolla una aplicación móvil empleando las herramientas y lenguajes que resuelvan un modelo de negocios de una problemática del mundo real.	3.3.1	▲	▲	▲	10%		
% PESO PARA LA UNIDAD						30%		
PESO TOTAL DEL MÓDULO						100%		

9. Desarrollo de actividades de evaluación.

Unidad de Aprendizaje:	1. Administración del ciclo de desarrollo de aplicaciones
Resultado de Aprendizaje:	1.1 Identifica ambientes de desarrollo empleando herramientas Android dentro del lenguaje de programación Java para el desarrollo eficiente de aplicaciones móviles
Actividad de Evaluación:	1.1.1. Desarrolla una aplicación básica para Android utilizando una herramienta de desarrollo visual

El diseño de la interfaz genera ventanas de actividades con sus componentes XML y Java
Los controles y eventos generan código para definir comportamiento de eventos en controles gráficos
El reporte escrito presenta el documento considerando:

- La estructura de contenidos solicitada
- Reglas ortográficas
- Glosario de términos
- Carátula e índice

Unidad de Aprendizaje:	1. Administración del ciclo de desarrollo de aplicaciones
Resultado de Aprendizaje:	1.2 Utiliza estándares de administración de proyectos utilizando el modelo de programación ágil para la optimización del ciclo de vida de desarrollo de software con el fin de aumentar la calidad del proceso y del producto en el desarrollo de aplicaciones
Actividad de Evaluación:	1.2.1. Desarrolla una aplicación prototípica conforme a la metodología SCRUM

Define proyecto de software considerando:

- Actores SCRUM
- Historias de usuario
- Backlog de funcionalidad

Ejecuta sprints de desarrollo de la aplicación según la metodología SCRUM

Presenta el documento escrito considerando:

- La estructura de contenidos solicitada
- Reglas ortográficas
- Glosario de términos
- Carátula e índice

Unidad de Aprendizaje:	2. Programación de la interfaz de la aplicación
Resultado de Aprendizaje:	2.1 Utiliza las estructuras Activity e Intent del lenguaje Java para la creación de la estructura de navegación de una aplicación móvil para el desarrollo eficiente de aplicaciones móviles
Actividad de Evaluación:	2.1.1. Construye una aplicación móvil para Android considerando: <ul style="list-style-type: none">• Diseño de <i>Activities</i>• Controles gráficos en <i>Activities</i>• Manejo de eventos• Código Java para respuesta a eventos en <i>Activities</i>• Uso de <i>Intent</i> para comunicación entre <i>Activities</i>

Redacta código de actividades con Layouts lineares y relativos.

Redacta código para la creación de Intents que enlacen Activities, con el componente Extra requerido para el paso de parámetros.

Presenta el documento escrito considerando:

- La estructura de contenidos solicitada
- Reglas ortográficas
- Glosario de términos
- Carátula e índice

Unidad de Aprendizaje:	2. Programación de la interfaz de la aplicación
Resultado de Aprendizaje:	2.2 Integra estructuras de Layout y Fragmentos en Java para crear interfaces fluidas empleando sus herramientas que permitan el despliegue de datos de forma variable en una aplicación
Actividad de Evaluación:	2.2.1. Desarrolla una aplicación Android con despliegue de datos variable por medio del uso de Layouts y fragmentos.

Modifica la estructura de la actividad para utilizar Layouts de contenido variable con al menos una de las siguientes:

- GridView
- ListView
- RecyclerView

Declara fragmentos por medio de Layouts de Tarjeta o de elemento de Lista.

Genera adaptadores para el relleno de fragmentos

Presenta el documento escrito considerando:

- La estructura de contenidos solicitada
- Reglas ortográficas
- Glosario de términos
- Carátula e índice

Unidad de Aprendizaje:	3. Programación de interacciones con datos externos
Resultado de Aprendizaje:	3.1 Utiliza elementos del lenguaje Java empleando sus herramientas para crear aplicaciones que interactúen con elementos multimedia y bases de datos locales.
Actividad de Evaluación:	3.1.1. Construye una aplicación Android considerando: <ul style="list-style-type: none">• Interacción con contenido multimedia• Acceso a bases de datos locales

Genera aplicaciones que consumen recursos internos como audio, video o imágenes.

Escribe clases para la administración de bases de datos SQLite

Presenta el documento escrito considerando:

- La estructura de contenidos solicitada
- Reglas ortográficas
- Glosario de términos
- Carátula e índice

Unidad de Aprendizaje:	3. Programación de interacciones con datos externos
Resultado de Aprendizaje:	3.2 Utiliza componentes del lenguaje Java conforme sus procedimientos para la creación de aplicaciones que consuman datos externos por medio de servicios web.
Actividad de Evaluación:	3.2.1. Construye una aplicación web considerando: <ul style="list-style-type: none">• Interacción con web services• Lectura de documentos XML y json• Envío de datos a web services

Genera código para la interpretación de cadenas XML y json

Genera código para la apertura de un stream de datos desde un web service

Genera código para la comunicación con un dispositivo externo

Presenta el documento escrito considerando:

- La estructura de contenidos solicitada
- Reglas ortográficas
- Glosario de términos
- Carátula e índice

Unidad de Aprendizaje:	3. Programación de interacciones con datos externos
Resultado de Aprendizaje:	3.3 Desarrolla una aplicación móvil empleando las herramientas y lenguajes que resuelvan un modelo de negocios de una problemática del mundo real.
Actividad de Evaluación:	3.3.1. Desarrollo de un proyecto integrador considerando: <ul style="list-style-type: none">• Aplicaciones con actividades múltiples• Manejo de eventos• Uso de contenido multimedia• Uso de contenido variable• Almacenamiento de datos• Uso de servicios Web• Administración de proyectos de TI

Identifica los roles SCRUM en el equipo

Genera tarjetas con historias del usuario SCRUM

Genera Backlog de actividades

Desarrolla la aplicación dentro del marco de sprints de SCRUM

Presenta el documento escrito considerando:

- La estructura de contenidos solicitada
- Reglas ortográficas
- Glosario de términos
- Carátula e índice

10. Matriz de valoración o rúbrica

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema:	DDAM-00	Nombre del módulo:	Diseño y desarrollo de aplicaciones móviles	Nombre del alumno:	
Docente evaluador:		Grupo:		Fecha:	
Resultado de aprendizaje:	1.1. Identifica ambientes de desarrollo empleando herramientas Android dentro del lenguaje de programación Java para el desarrollo eficiente de aplicaciones móviles		Actividad de evaluación:	1.1.1. Desarrolla una aplicación básica para Android utilizando una herramienta de desarrollo visual	

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Diseño de interfaz	45%	Genera ventanas de actividades con sus componentes XML y Java Además presenta Layouts adicionales.	Genera ventanas de actividades con sus componentes XML y Java	Genera ventanas de actividades omitiendo alguno de los siguientes elementos <ul style="list-style-type: none"> • Diseño de componente visual XML • Programación de comportamientos Java
Elaboración de controles y eventos	45%	Genera código para definir comportamiento de eventos en controles gráficos. Además describe técnicas para asignar una función de comportamiento a un control.	Genera código para definir comportamiento de eventos en controles gráficos	No presenta comportamientos asignados a controles

<p>Reporte escrito</p>	<p>10%</p>	<p>Presenta el documento considerando:</p> <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice <p>Incluye imágenes ilustrativas y utiliza lenguaje técnico.</p> <p>El documento es presentado con carátula, índice, numeración de páginas y con presentación profesional.</p>	<p>Presenta el documento considerando:</p> <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice 	<p>El documento omite o aborda incorrectamente alguno de los siguientes elementos:</p> <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice
<p>100</p>				

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema:	DDAM-00	Nombre del módulo:	Diseño y desarrollo de aplicaciones móviles	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	1.2. Utiliza estándares de administración de proyectos utilizando el modelo de programación ágil para la optimización del ciclo de vida de desarrollo de software con el fin de aumentar la calidad del proceso y del producto en el desarrollo de aplicaciones			Actividad de evaluación:	1.2.1. Desarrolla una aplicación prototípica conforme a la metodología SCRUM

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Definición del proyecto de software	45%	Define proyecto de software considerando: <ul style="list-style-type: none"> Actores SCRUM Historias de usuario Backlog de funcionalidad Además, evalúa la eficiencia de la metodología en situaciones del ambiente laboral real	Define proyecto de software considerando: <ul style="list-style-type: none"> Actores SCRUM Historias de usuario Backlog de funcionalidad 	Omite alguno de los siguientes elementos en el proyecto de software: <ul style="list-style-type: none"> Actores SCRUM Historias de usuario Backlog de funcionalidad
Ejecución del proyecto de software	45%	Ejecuta sprints de desarrollo de la aplicación según la metodología. Además, evalúa la calidad del proceso de software	Ejecuta sprints de desarrollo de la aplicación según la metodología SCRUM	No sigue la metodología SCRUM en el desarrollo del app
Reporte escrito	10%	Presenta el documento considerando: <ul style="list-style-type: none"> La estructura de contenidos solicitada Reglas ortográficas 	Presenta el documento considerando: <ul style="list-style-type: none"> La estructura de contenidos solicitada 	El documento omite o aborda incorrectamente alguno de los siguientes elementos:

		<ul style="list-style-type: none"> • Glosario de términos • Carátula e índice <p>Incluye imágenes ilustrativas y utiliza lenguaje técnico. El documento es presentado con carátula, índice, numeración de páginas y con presentación profesional.</p>	<ul style="list-style-type: none"> • Reglas ortográficas • Glosario de términos • Carátula e índice 	<ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice
	100%			

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema:	DDAM-00	Nombre del módulo:	Diseño y desarrollo de aplicaciones móviles	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:		2.1.Utiliza las estructuras Activity e Intent del lenguaje Java para la creación de la estructura de navegación de una aplicación móvil para el desarrollo eficiente de aplicaciones móviles		Actividad de evaluación:	<p>2.1.1. Construye una aplicación móvil para Android considerando:</p> <ul style="list-style-type: none"> • Diseño de <i>Activities</i> • Controles gráficos en <i>Activities</i> • Manejo de eventos • Código Java para respuesta a eventos en <i>Activities</i> • Uso de <i>Intent</i> para comunicación entre <i>Activities</i>

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Definición de Activities	45%	Redacta código de actividades con Layouts lineares y relativos. Además describe las actividades en función de requerimientos funcionales de usuario.	Redacta código de actividades con Layouts lineares y relativos.	No demuestra dominio en el uso de layouts lineares o relativos
Definición de Intents	45%	Redacta código para la creación de Intents que enlacen Activities, con el componente Extra requerido para el paso de parámetros. Adicionalmente realiza documentación del código con estándares de documentación ágiles	Redacta código para la creación de Intents que enlacen Activities, con el componente Extra requerido para el paso de parámetros.	<ul style="list-style-type: none"> • Omite el código de Intent • Omite el paso de parámetros en el componente Extra • Omite la recuperación de parámetros en el componente Extra

Reporte escrito	10%	<p>Presenta el documento considerando:</p> <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice <p>Incluye imágenes ilustrativas y utiliza lenguaje técnico.</p> <p>El documento es presentado con carátula, índice, numeración de páginas y con presentación profesional.</p>	<p>Presenta el documento considerando:</p> <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice 	<p>El documento omite o aborda incorrectamente alguno de los siguientes elementos:</p> <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice
	100%			

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema:	DDAM-00	Nombre del módulo:	Diseño y desarrollo de aplicaciones móviles	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	2.2. Integra estructuras de Layout y Fragmentos en Java para crear interfaces fluidas empleando sus herramientas que permitan el despliegue de datos de forma variable en una aplicación			Actividad de evaluación:	2.2.1. Desarrolla una aplicación Android con despliegue de datos variable por medio del uso de Layouts y fragmentos.

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Declaración de Layouts variables	45%	Modifica la estructura de la actividad para utilizar Layouts de contenido variable con más de una de las siguientes: <ul style="list-style-type: none"> • GridView • ListView • RecyclerView 	Modifica la estructura de la actividad para utilizar Layouts de contenido variable con al menos una de las siguientes: <ul style="list-style-type: none"> • GridView • ListView • RecyclerView 	No modifica la estructura de la actividad con Layouts variables
Declaración de fragmentos	45%	Declara fragmentos por medio de Layouts de Tarjeta o de elemento de Lista. Además Declara fragmentos con: <ul style="list-style-type: none"> • ListItem • CardView 	Declara fragmentos por medio de Layouts de Tarjeta o de elemento de Lista. Genera adaptadores para el relleno de fragmentos	Omite alguna de las siguientes: <ul style="list-style-type: none"> • Declaración del Layout de Fragmento • Declaración del Adaptador
Reporte escrito	10%	Presenta el documento considerando: <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos 	Presenta el documento considerando: <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas 	El documento omite o aborda incorrectamente alguno de los siguientes elementos: <ul style="list-style-type: none"> • La estructura de contenidos solicitada

		<ul style="list-style-type: none">• Carátula e índice Incluye imágenes ilustrativas y utiliza lenguaje técnico. El documento es presentado con carátula, índice, numeración de páginas y con presentación profesional.	<ul style="list-style-type: none">• Glosario de términos• Carátula e índice	<ul style="list-style-type: none">• Reglas ortográficas• Glosario de términos• Carátula e índice
	100			

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema:	DDAM-00	Nombre del módulo:	Diseño y desarrollo de aplicaciones móviles	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	3.1 Utiliza elementos del lenguaje Java empleando sus herramientas para crear aplicaciones que interactúen con elementos multimedia y bases de datos locales.		Actividad de evaluación:	3.1.1. Construye una aplicación Android considerando: <ul style="list-style-type: none"> • Interacción con contenido multimedia • Acceso a bases de datos locales 	

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Manejo de recursos internos	45%	Genera aplicaciones que consumen recursos internos como audio, video o imágenes. De manera adicional genera animaciones avanzadas con los recursos	Genera aplicaciones que consumen recursos internos como audio, video o imágenes.	El código no puede acceder a los recursos internos
Manejo de bases de datos	45%	Escribe clases para la administración de bases de datos SQLite Adicionalmente incluye varias tablas.	Escribe clases para la administración de bases de datos SQLite	<ul style="list-style-type: none"> • Omite el código de la clase de acceso a SQLite • Genera instrucciones SQL erróneas
Reporte escrito	10%	Presenta el documento considerando: <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice 	Presenta el documento considerando: <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice 	El documento omite o aborda incorrectamente alguno de los siguientes elementos: <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice

		Incluye imágenes ilustrativas y utiliza lenguaje técnico. El documento es presentado con carátula, índice, numeración de páginas y con presentación profesional.		
	100%			

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema:	DDAM-00	Nombre del módulo:	Diseño y desarrollo de aplicaciones móviles	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:	3.2. Utiliza componentes del lenguaje Java conforme sus procedimientos para la creación de aplicaciones que consuman datos externos por medio de servicios web.		Actividad de evaluación:	3.2.1. Construye una aplicación web considerando: <ul style="list-style-type: none"> • Interacción con web services • Lectura de documentos XML y json • Envío de datos a web services 	

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Codificación de datos	45%	Genera código para la interpretación de cadenas XML y json con estructuras complejas	Genera código para la interpretación de cadenas XML y json	No consigue interpretar cadenas json o XML
Servicios web	45%	<ul style="list-style-type: none"> • Genera código para la apertura de un stream de datos desde un web service • Genera código para la comunicación con un dispositivo externo • Genera código para interactuar con componentes de internet de las cosas 	<ul style="list-style-type: none"> • Genera código para la apertura de un stream de datos desde un web service • Genera código para la comunicación con un dispositivo externo 	<ul style="list-style-type: none"> • Omite el código para la apertura de un stream de datos desde un web service • Omite el código para la comunicación con un dispositivo externo
Reporte escrito	10%	Presenta el documento considerando: <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos 	Presenta el documento considerando: <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos 	El documento omite o aborda incorrectamente alguno de los siguientes elementos: <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos

		<ul style="list-style-type: none">• Carátula e índice Incluye imágenes ilustrativas y utiliza lenguaje técnico. El documento es presentado con carátula, índice, numeración de páginas y con presentación profesional.	<ul style="list-style-type: none">• Carátula e índice	<ul style="list-style-type: none">• Carátula e índice
	100%			

MATRIZ DE VALORACIÓN O RÚBRICA

Siglema:	DDAM-00	Nombre del módulo:	Diseño y desarrollo de aplicaciones móviles	Nombre del alumno:	
Docente evaluador:				Grupo:	Fecha:
Resultado de aprendizaje:		3.3. Desarrolla una aplicación móvil empleando las herramientas y lenguajes que resuelvan un modelo de negocios de una problemática del mundo real.		Actividad de evaluación:	3.3.1. Desarrollo de un proyecto integrador considerando: <ul style="list-style-type: none"> • Aplicaciones con actividades múltiples • Manejo de eventos • Uso de contenido multimedia • Uso de contenido variable • Almacenamiento de datos • Uso de servicios Web • Administración de proyectos de TI

INDICADORES	%	CRITERIOS		
		Excelente	Suficiente	Insuficiente
Planeación de la aplicación	45%	<ul style="list-style-type: none"> • Identifica los roles SCRUM en el equipo • Genera tarjetas con historias del usuario SCRUM • Genera Backlog de actividades • De manera adicional agrega documentación del sistema según estándares de desarrollo ágiles 	<ul style="list-style-type: none"> • Identifica los roles SCRUM en el equipo • Genera tarjetas con historias del usuario SCRUM • Genera Backlog de actividades 	<ul style="list-style-type: none"> • No identifica los roles SCRUM en el equipo • No genera tarjetas con historias del usuario SCRUM • No genera Backlog de actividades
Desarrollo de la aplicación	45%	Desarrolla la aplicación dentro del marco de sprints de SCRUM De manera adicional la aplicación resuelve un problema de interés para un sector económico regional	Desarrolla la aplicación dentro del marco de sprints de SCRUM	<ul style="list-style-type: none"> • No presenta evidencia de disciplina SCRUM • No presenta una aplicación funcional

<p>Reporte escrito</p>	<p>10%</p>	<p>Presenta el documento considerando:</p> <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice <p>Incluye imágenes ilustrativas y utiliza lenguaje técnico. El documento es presentado con carátula, índice, numeración de páginas y con presentación profesional.</p>	<p>Presenta el documento considerando:</p> <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice 	<p>El documento omite o aborda incorrectamente alguno de los siguientes elementos:</p> <ul style="list-style-type: none"> • La estructura de contenidos solicitada • Reglas ortográficas • Glosario de términos • Carátula e índice
<p>100</p>				