

**UNIVERSIDAD AUTÓNOMA DE CHIAPAS
LICENCIATURA EN SISTEMAS COMPUTACIONALES**

Área de formación: Elección Libre
Unidad académica: Programación de dispositivos móviles con Java
Ubicación: Noveno Semestre.
Clave: 2111
Horas semana-mes: 4
Horas teoría: 1
Horas práctica: 3
Unidades CONAIC: 32
Prerrequisitos: Ninguno
Horas de infraestructura: 3
Créditos: 5

PRESENTACIÓN

El diseño de Java, su robustez, el respaldo de la industria y su fácil portabilidad lo han hecho uno de los lenguajes con un mayor crecimiento en distintos ámbitos de la industria de la informática.

Desde la creación de la especificación J2ME (Java 2 Platform, Micro Edition) y la facilidad de adquisición de microprocesadores específicamente diseñados para ejecutar bytecode Java y software Java para tarjetas inteligentes (JavaCard) como teléfonos móviles, buscapersonas, set-top-boxes, sintonizadores de TV y otros pequeños electrodomésticos, han producido toda una revolución en el mercado de dispositivos electrónicos de consumo.

OBJETIVO GENERAL

El alumno empleará sus habilidades en base a sus conocimientos en la programación orientada a objetos a través del lenguaje de programación Java para el Desarrollo de aplicaciones de telefonía móvil utilizando la plataforma J2ME (Java 2, Micro Edition), la tecnología CDC (Configuración de dispositivo conectada) y el perfil MIDP (Perfil de dispositivo de información móvil).

UNIDAD I.- PANORAMICA DE LA TECNOLOGÍA J2ME

TIEMPO APROXIMADO: 3 hrs.

OBJETIVO DE LA UNIDAD:

El alumno conocerá que J2ME es una versión minimizada de la plataforma Java 2, orientada a dispositivos móviles debido a que éstos tienen una potencia de cálculo baja e interfaces de usuario pobres que no necesitan todo el soporte que brinda el J2SE.

CONTENIDO

- 1.1. Arquitectura J2ME
 - 1.1.1 Consideraciones de la plataforma
 - 1.1.2 Dispositivos que usan la tecnología J2ME
 - 1.1.3 Capas de software para la tecnología J2ME
 - 1.1.4 Implementación del CLDC y CLDC-HI
 - 1.1.5 Extensión de la funcionalidad del CLDC con perfiles.

UNIDAD II.- APLICACIONES MIDP (Mobile Information Device Profile)

TIEMPO APROXIMADO: 3hrs.

OBJETIVO DE LA UNIDAD:

- 2.1. Medioambiente de las aplicaciones MIDP 2.0
 - 2.1.1 Configuración de las plataformas que soportan MIDP
 - 2.1.2 Especificación de MIDP 2.0
 - 2.1.3 Software de administración de la aplicación
- 2.2. Proceso de desarrollo con MIDP
 - 2.2.1 Paquete MIDlet
 - 2.2.2 Recursos del MIDlet
 - 2.2.3 Ciclo de vida de una MIDlet
 - 2.2.4 Salida de un MIDlet

UNIDAD III.- INTERFACES DE ALTO NIVEL

TIEMPO APROXIMADO: 5hrs.

OBJETIVO DE LA UNIDAD:

- 3.1. Arquitectura de interfaz de usuario del MIDP
 - 3.1.1 Resumen del API a bajo y alto nivel
 - 3.1.2 Paquete MIDP lcdui
 - 3.1.3 Jerarquía de clases del API de alto nivel LCDUI
 - 3.1.4 Interfaces de alto nivel del API
- 3.2. Clases
 - 3.2.1 La clase Display
 - 3.2.2 La clase Displayable
 - 3.2.3 La clase Screen
 - 3.2.3.1 Diferentes tipos de Screens
 - 3.2.3.2 Mostrando un objeto Screen en el dispositivo
 - 3.2.3.3 Creación y despliegue de objetos Screen simples
 - 3.2.4 La clase Lista
 - 3.2.5 La clase TextBox
 - 3.2.6 La clase Alert
 - 3.2.7 La clase Form
 - 3.2.8 La clase Item
 - 3.2.8.1 Tipos de clases Item
 - 3.2.8.1.1 La clase StringItem
 - 3.2.8.1.2 La clase Image
 - 3.2.8.1.3 La clase ImageItem
 - 3.2.9 Desplegando un mensaje Ticker
 - 3.2.10 El objeto Spacer
 - 3.2.11 Elementos editables por el usuario
 - 3.2.12 La clase TextField
 - 3.2.12.1 Creación y uso de la clase TextField
 - 3.2.12.2 La clase DateField
 - 3.2.12.3 Creación y uso de objetos DateField
 - 3.2.13 La clase Gauge
 - 3.2.13.1 Creación y uso de objetos Gauge

- 3.2.14 La interfaz Choice
 - 3.2.14.1 La clase ChoiceGroup
 - 3.2.14.2 Creación de objetos ChoiceGroup
- 3.2.15 Manejo de Eventos Item de Alto Nivel
 - 3.2.15.1 Manejando eventos de estado para un Item
 - 3.2.15.2 Creación de un manejador de eventos
 - 3.2.15.3 Registrando un manejador de eventos
- 3.2.16 El objeto Command
 - 3.2.16.1 La interfaz CommandListener
 - 3.2.16.2 Uso de objetos de tipo Comando
 - 3.2.16.3 Creando un objeto de tipo Comando
 - 3.2.16.4 Manejo de objetos de tipo Comando
- 3.2.17 Implementar la interfaz CommandListener
 - 3.2.17.1 Registrar el CommandListener con la Forma o el Item

UNIDAD IV.- INTERFACES DE USUARIO DE BAJO NIVEL

TIEMPO APROXIMADO: 5 hrs.

OBJETIVO DE LA UNIDAD:

- 4.1 El API UI MIDP de Bajo Nivel
 - 4.2 La clase Canvas
 - 4.2.1 Extendiendo la clase Canvas
 - 4.2.2 Desplegando un objeto Canvas en un MIDlet
 - 4.2.3 Usando la clase Graphics
 - 4.2.4 Subclases a partir de la clase Canvas
 - 4.2.5 Desplegando un objeto Canvas
 - 4.3 Modelo de pintado
 - 4.3.1 Metodos de pintado
 - 4.4 Manejo de Eventos para la UI de bajo nivel
 - 4.5 Teclados de dispositivos móviles
 - 4.5.1 Manejo de eventos basados en teclado
 - 4.6 Definiendo los modos Keycode y GameCode
 - 4.6.1 Convirtiendo los KeyCodes a GameCodes
 - 4.6.2 Manejo de eventos de teclado
 - 4.6.3 Manejando eventos de apuntadores

UNIDAD V.- EXTENSION DE INTERFACES DE ALTO Y BAJO NIVEL

TIEMPO APROXIMADO: 5 hrs.

OBJETIVO DE LA UNIDAD:

- 5.1 La clase CustomItem
 - 5.1.1 Los conceptos de la clase CustomItem
 - 5.1.2 Implementando la clase CustomItem
 - 5.1.3 Implementando los metodos abstractos

- 5.1.4 Implementando el metodo Paint
- 5.1.5 Manejo de eventos de objetos CustomItem
- 5.1.6 La notificacion de cambio de objetos de tipo CustomItem
- 5.2 Sobreescritura de metodos keyXXX()
- 5.3 Sobreescritura de metodos pointerXXX()
- 5.4 El paquete Game
 - 5.4.1 La clase GameCanvas
- 5.5 La clase Layer
 - 5.5.1 Usando una clase TiledLayer para crear el fondo
- 5.6 La clase Sprite
 - 5.6.1 Creación de un objeto Sprite para usar en el Canvas del juego
 - 5.6.2 Detectando colisiones entre los objetos Sprite
- 5.7 Peleando por el teclado

UNIDAD VI.- MANEJO DE DATOS EN EL DISPOSITIVO

TIEMPO APROXIMADO: 5hrs.

OBJETIVO DE LA UNIDAD:

- 6.1 Datos locales y el MIDP
- 6.2 El paquete RMS
 - 6.2.1 Accediendo a RecordStores en el dispositivo
 - 6.2.2 Creación y apertura de un objeto de tipo RecordStore
 - 6.2.3 Recuperación de información de un objeto de tipo RecordStore
 - 6.2.4 Recuperando información de registros.
 - 6.2.5 Cerrando el objeto RecordStore
 - 6.2.6 Almacenamiento y recuperación de datos en un objeto RecordStore.
- 6.3 Recuperar un solo registro
 - 6.3.1 Agregar un registro
 - 6.3.2 Actualizar un registro
 - 6.3.3 Borrar un registro
- 6.4 Usando la interfaz RecordEnumeration
 - 6.4.1 Listando Objetos de tipo RecordStore
 - 6.4.2 Removiendo un objeto RecordStore
 - 6.4.3 Relación entre instancias RecordStore y MIDlets
 - 6.4.4 Compartiendo un RecordStores entre varios MIDlets
- 7.1 Accediendo a Servicios y Datos de Red.
 - 7.1.1 Accediendo a datos con el GCF
 - 7.1.2 Interfaces GCF
 - 7.1.3 La clase Connector
- 7.2 Los formatos URL
- 7.3 Protocolos soportados por GCF
 - 7.3.1 Usando la fábrica Connection
 - 7.3.2 Usando las interfaces GCF
- 7.4 Intercambiando datos sobre HTTP
 - 7.4.1 Estados de las conexiones HTTP
- 7.5 Aplicando el contenido de la información
 - 7.5.1 Usando la interfazHttp Connection
 - 7.5.2 Configuración y recuperación de información especifica del HTTP
- 7.5.3 Enviando y recibiendo datos usando los métodos GET y SEND del HTT

- 8.1 Control de medios.
 - 8.1.1 Resumen de medios.
 - 8.1.2 El paquete
- 8.2 MMAPAPI
 - 8.2.1 La interfaz MMAPAPI
 - 8.2.2 Las clases Manager y Player
 - 8.2.3 Los estados de la clase Player
- 8.3 Recuperando Audio y Video como un recurso local
 - 8.3.1 Generando tonos simples
 - 8.3.2 Generando una secuencia de tonos

- 9.1 Seguridad en MIDP
 - 9.1.1 La seguridad y el MIDP 2.0
 - 9.1.2 Resumen de seguridad
 - 9.1.3 Conjuntos confiables e inconfiables
 - 9.1.4 Dominio de protección
 - 9.1.5 Conjuntos confiables
 - 9.1.6 Comunicaciones de red seguras
- 9.2 Tecnología PUSH
 - 9.2.1 Conexiones dentro de la frontera
 - 9.2.2 Determinando la forma de ejecución de un MIDlet
 - 9.2.3 Calendarizado la ejecución de un MIDlet

- 10.1 Aprovechamiento sobre la marcha
 - 10.1.1 Descargando aplicaciones J2ME
 - 10.1.2 Instalación de aplicaciones
 - 10.1.3 Requerimientos funcionales OTA
- 10.2 Proceso de instalación de aplicaciones
- 10.3 Proceso de búsqueda de conjuntos
 - 10.3.1 Proceso de instalación de conjuntos
 - 10.3.2 Posibles errores de instalación
- 10.4 Actualización de conjuntos
 - 10.4.1 Proceso de ejecución de conjuntos
 - 10.4.2 Proceso de borrado de conjuntos

EXPERIENCIA DE APRENDIZAJE.

- Uso de portal en Internet que cuente con un foro de discusión
- Utilizar software didáctico y de apoyo
- Ejercicios grupales fuera y dentro de clase.
- Formar equipos de trabajo para la exposición de investigaciones y tareas
- Fomentar el trabajo en equipo
- Creación de programas
- Solución de problemas
- Presentación de un proyecto final

CRITERIOS DE EVALUACIÓN

Exámenes por unidad	50%
Investigación y tareas	10%

Programas por computadora	20%
Proyecto Final	<u>20%</u>
TOTAL	100%

RECURSOS NECESARIOS

Recursos tecnológicos:

Windows, Java, Netbeans.

PRÁCTICAS SUGERIDAS

- Desarrollar aplicaciones para entender el ciclo de vida de un programa para celulares.
- Desarrollo de programas que en la que se usen interfaces de alto y bajo nivel controlando el teclado con las clases apropiadas.
- Desarrollar aplicaciones que almacenen datos persistentes en el celular.
- Desarrollo de un videojuego surgido de la inventiva de los alumnos.

BIBLIOGRAFÍA BÁSICA

FRIESEN, J.(2010). Learn Java for Android Development. USA: Apress.

TREFRY, G. (2010). Casual game design: designing play for the gamer in all of us. USA: Morgan Kaufmann.

BIBLIOGRAFÍA COMPLEMENTARIA

KING, C. (2009). Advanced BlackBerry development. USA: Apress.

CONDER, S. (2010). Android Wireless Application Development. USA: Addison-Wesley.