

# Máster Internacional en Programación de Aplicaciones y Dispositivos Móviles

## **CARGA LECTIVA: 1.400 HORAS**

### **Formación técnica y Certificación: 400 h.**

El alumno realiza la formación técnica utilizando las últimas tecnologías de formación abierta interactuando con un completo contenido de material multimedia que combinan la teoría, las prácticas y módulos de evaluación. El alumno cuenta con el apoyo de un tutor en línea que de forma presencial e individualmente le va explicando los contenidos y le ayuda a realizar las prácticas. El alumno también prepara en la parte final del curso los distintos exámenes de certificación antes de presentarse a los exámenes oficiales.

### **Formación específica. Clases: 480 h.**

Una vez el alumno alcanza un determinado nivel en su formación técnica, comienza según los horarios programados el cuerpo principal de la profesión. Esta parte del curso se imparte en formación presencial o telepresencial con clases en directo a través de internet que quedan grabadas para que el alumno pueda recuperar o utilizar como material de estudio. El alumno completa esta formación que impartimos en grupos reducidos con prácticas en nuestros laboratorios que se pueden extender fuera del horario de clase y están coordinados por su profesor.

### **Proyectos: 120 h.**

Esta parte del curso forma parte de la preparación de nuestros alumnos para la formación dual de la empresa que los acogerá en prácticas. A través de los proyectos realizados en nuestras aulas, coordinados por su tutor el alumno con los conocimientos que acumuló en clase aprende la metodología a seguir para realizar un trabajo profesional, utilizando una actitud proactiva para resolver problemas, organizar las tareas y cerrar los presupuestos de ejecución.

### **Práctica en empresas: 400 h.**

Una vez el alumno ha finalizado la actividad de proyectos, ha superado las evaluaciones y se ha certificado, está listo para incorporarse a una empresa que lo acoge en prácticas. El alumno bajo la dirección de PC Carrier se dirige a empresas y organismos que colaboran en la formación dual de los alumnos, completando su formación para poder iniciar a continuación una relación laboral en la categoría profesional en la que se está formando.

### **London Calling:**

Formación Alojamiento y Servicios en Inglaterra (5 meses)

Incluye:

- . Programa Doble Inglés C1/Advanced + Programa Iniciación Alemán
- . Prácticas Empresas Inglaterra + Bolsa Empleo en España y Europa
- . Becas Horizonte 2020 + Módulos curriculares que demanda la empresa receptora

# **Módulo 1: Especialidad Programador Java y Dispositivos Android**

## **1.1 PROGRAMADOR JAVA DE ORACLE**

### **OBJETIVO**

Adquirir los conocimientos de un programador Java de Oracle, para trabajar en distintos entornos de programación, y la forma de implantar aplicaciones en entornos Windows y Web.

### **REQUISITOS PREVIOS**

Conocimientos básicos de informática e Internet.

### **CERTIFICACIÓN**

Oracle Certified Professional, Java SE 7 Programmer  
Java EE 5 Web Component Developer Certified Professional

### **FORMACIÓN TÉCNICA Y CERTIFICACIÓN**

#### **ORACLE CERTIFIED ASSOCIATE, JAVA SE 7 PROGRAMMER I. Examen 1Z0-803**

Java Basics. Tipos de datos Java. Datos usando la clase String-Builder. Crear y manipular Strings. Operadores. Construcciones de decisión. Arrays. Construcciones de repetición. Métodos y encapsulación. Herencia. Polimorfismo. Gestión de excepciones.

#### **ORACLE DEVELOPER. PL/SQL**

Operaciones DML, DDL y DCL. Diseño y creación de BBDD. Objetos de base de datos. Bloques PL/SQL. Funciones y procedimientos almacenados. Transacciones. Triggers. Datos CLOB y BLOB. SQL dinámico. Modelo objeto-relacional.

### **FORMACIÓN ESPECÍFICA**

#### **ORACLE CERTIFIED PROFESSIONAL, JAVA SE 7 PROGRAMMER II. Examen 1Z0-804**

Diseño de clases Java. Sobrescribir métodos. Sobrecarga de constructores y métodos. Clases abstractas y subclasses. Herencia de interfaz y herencia de clase. Genéricos y colecciones. Clases wrapper, autoboxing y unboxing. Implementaciones List, Set y Deque. Implementaciones Map. Usar java.util. Comparator y java.util. Comparable. Procesar Strings. Scanner, StringTokenizer, StringBuilder, String y Formatter. Patrones de expresión y parámetros de formato. Excepciones y Assertions. Interfaz AutoCloseable. Paquete java.io. Java File I/O (NIO.2). Clases Path y Files. Interfaces BasicFileAttributes, DosFileAttributes, PosixFileAttributes. Aplicaciones de acceso a datos con JDBC. Hilos de ejecución. Clase Thread, Interfaz Runnable. Concurrencia. Paquete java.util.concurrent. Clases Lock, ReadWriteLock y ReentrantLock del paquete java.util.concurrent.locks. Localización. Objeto Locale.

## **JAVA EE 5 WEB COMPONENT DEVELOPER CERTIFIED PROFESSIONAL EXAM. Examen 1Z0-858**

Modelo de tecnología Servlet. Métodos HTTP, protocolos. Cookies. Interfaz Http Servlet Response. Páginas JSP. Clases servlet. Tag Libraries. Archivos JAR. Archivos de clase Java. Protección de recursos de accesos HTTP. Archivo WAR. Modelo Web Container. Modelo de proceso web container request. Gestión de sesiones. APIs de acceso al objeto de sesión. Autenticación, autorización, integridad de datos, confidencialidad. Restricciones de seguridad. Configuración de login. Modelo Java-Server Pages (JSP). Elementos del lenguaje. Directivas. JSP Document. Eventos y objetos. Expression Language (EL). Standard Actions. Tag Libraries. Patrones Java EE. Model-View-Controller.

## **JAVA EE 5 BUSINESS COMPONENT DEVELOPER & WEB SERVICES DEVELOPER**

EJB 3.0 Enterprise Bean. API de entidades de Persistence. Operaciones de entidad. Unidades y contextos de Persistence. Java Persistence Language Query. Tecnologías de acceso remoto. Transacciones y Excepciones. Conceptos de seguridad. Servicios Web XML estándar. Servicios Web SOAP 1.2. Servicios Web J2EE. Publicación. Seguridad. Desarrollo de servicios web. Clientes de servicios Web.

## **1.2 Android Application Engineer**

### **OBJETIVO**

Aprender a trabajar con los componentes y librerías proporcionadas por Android para desarrollar aplicaciones que se ejecutarán en dispositivos móviles de última generación que tengan instalado este sistema operativo.

### **REQUISITOS PREVIOS**

Conocimientos de Java. Se recomienda haber realizado con anterioridad o estar realizando el curso de Programador Java.

### **CERTIFICACIÓN**

OESF Certified Engineer for Android™ Application Basic Level (ACE)

### **FORMACIÓN TÉCNICA**

Introducción a la plataforma Android.  
Fundamentos de programación Android.

### **FORMACIÓN ESPECÍFICA**

#### **ANDROID FRAMEWORK**

Arquitectura del sistema. Licencias. Interfaz de usuario. Menús. Opciones de menú y menús de diálogo. Diálogos. Procesado de eventos. Eventos de botón, de teclado y de movimiento. Notificaciones. Barra de notificaciones, Clock Display. Estilos y temas. Vistas. Objeto vista. Gráficos 2D y 3D. Dibujar con Canvas y OpenGL. Archivos de recursos. Manifiesto Android. Nombre de la aplicación, método de ejecución, definición de Actividad principal. Formatos XML. Alias de recursos e internacionalización. Sustitución de recursos mediante la opción Alternative. Conceptos de seguridad, seguridad y permisos. Componentes de la aplicación. Intenciones y filtros de intenciones. Roles de intenciones. Vínculo con Applications. Intenciones implícitas y explícitas. Ciclo de vida de Actividades. Proveedores de contenidos. Ciclo de vida del servicio. BroadCast Receiver. Almacenamiento. Introducción a archivos I/O. Preferencias. Guardar y leer datos de aplicación. SQLite. Uso de SQL en la base de datos. Comunicaciones. Protocolos de red (HTTP, TCP/IP). Paquetes java.net y [android.net](http://android.net). Bluetooth. Paquete android.bluetooth. WiFi. Paquete WiFi de Android. Dispositivos y gestores externos. GPS. Paquete de sensor GPS. Sensores. Paquete sensor. Multimedia. Audio. Video. Utilizar archivos media. Testing. Framework Junit. Pruebas automatizadas. Código nativo. JNI. Lenguaje de script (ASE). Introducción a NDK.

## **PUBLICACIÓN DE LA APLICACIÓN**

Gestión de la aplicación. Versiones. Grabar la versión en el archivo de manifiesto. Generación de Clave Privada. Uso de las herramientas de clave para crear claves privadas. Clave de registro Map API. Uso de MapView. Distribución de la aplicación. Firma. Firma del archivo apk. Actualización. Actualización automática desde Android Market por versión.

## **FUNCIONES DE SOPORTE**

Funciones de desarrollo. Eclipse. Android Development Tools Plugin (adt). Diagnóstico de archivos con Layoutopt. Edición de imágenes con Draw 9-patch. Creación de archivos apk con Android Asset Packaging Tool (aapt). Definición de interfaces con diferentes capas con Android Interface Description Language (aidl). SQLite con sqlite3. Conversión de archivos de clase en archivos dex con dx. Optimización de archivos apk con zipalign. Funciones de depuración. Android Emulator. Android Virtual Devices (AVDs). Mostrar jerarquía de archivo con Hierarchy Viewer. Dalvik Debug Monitor Server (DDMS). Android Debug Bridge (adb). Tracer aplicaciones Android con Traceview. Crear una imagen de tarjeta SD con mksdcard. UI/Applications Exerciser Monkey. Comando android para gestión de AVD y SDK.

# **MÓDULO 2: Especialidad en Programación de Sistemas con Visual C# y Objective-C**

## **2.1 PROGRAMACIÓN VISUAL STUDIO.NET. MCSD**

### **OBJETIVO**

Desarrollo de aplicaciones para Windows 8 y Windows Store utilizando la plataforma Microsoft Visual Studio 2012. Desarrollo de aplicaciones web con ASP.NET, HTML5, CSS3 y JavaScript.

### **REQUISITOS PREVIOS**

Conocimientos básicos de informática, manejo de windows a nivel usuario, conocimientos básicos de metodología de programación.

### **CERTIFICACIÓN**

MCSD de Microsoft

### **FORMACIÓN TÉCNICA**

Microsoft SQL Server database Developer  
Metodología de la programación: Desarrollo Ágil (SCRUM)  
Microsoft SQL Server database Administrator

### **FORMACIÓN ESPECÍFICA**

#### **PROGRAMMING IN C#. Examen 483**

Sintaxis C#. Creación de métodos, manejar excepciones, monitorear aplicaciones. Desarrollo de código para una aplicación gráfica. Creación de clases e implementación de colecciones con seguridad de tipos y de jerarquías de clases usando herencia. Lectura y escritura de datos locales.

#### **ESSENTIALS OF DEVELOPING WINDOWS STORE APPS USING C#. Examen 484**

Introducción a la plataforma Windows 8 y a las aplicaciones Windows Store. Creación de interfaces de usuario usando XAML. Presentación de los datos. Trabajar con controles de presentación de datos. Implementar el diseño usando controles Windows 8 Built-in. Manejo de archivos en aplicaciones Windows 8.

#### **ADVANCED WINDOWS STORE APP DEVELOPMENT USING C#. Examen 485**

Básicos de aplicaciones. Implementar animaciones y transiciones, globalización y localización. Interfaces de usuario representativas y sin fisuras. Escenarios de datos avanzados en aplicaciones. Creación de componentes y controles reutilizables. Implementar escenarios avanzados de contratos.

## **PROGRAMMING IN HTML5 WITH JAVASCRIPT AND CSS3. Examen 480**

Introducción a HTML y CSS Creación de páginas de estilo HTML5. Introducción a JavaScript. Creación de formularios para recolectar datos y validar entradas de usuario. Comunicación con un origen de datos remoto. Estilos en HTML5 usando CSS3. Creación de objetos y métodos usando JavaScript. ASP .NET MVC4: Explorar. Diseño de aplicaciones web. Desarrollo de modelos, controladores y vistas. Testar y depurar aplicaciones web. Estructurar aplicaciones web. Aplicar estilos en aplicaciones web. Construir páginas de respuesta en aplicaciones web. Usar JavaScript y jQuery para páginas de respuesta en aplicaciones web. Controlar el acceso a datos en aplicaciones web. Introducción a servicios y tecnologías cloud. Consultas y manipulación de datos usando Entity Framework. ADO.NET. Creación y uso de ASP.NET Web API Services. Seguridad en ASP.NET Web API Services. Creación y diseño de servicios WCF.

## **2.2 PROGRAMACIÓN IPHONE & IPAD. OBJECTIVE-C**

### **OBJETIVOS**

Aprender a crear aplicaciones que se ejecutarán en dispositivos móviles Apple: entornos iPhone & iPad.

### **REQUISITOS**

Conocimientos de algún lenguaje del estilo C, y estar familiarizado con los conceptos de programación orientada a objetos.

### **FORMACIÓN TÉCNICA**

Introducción a Objective-C. Clases, objetos y métodos. Propiedades. Envío de mensajes. Foundation Classes: strings, arrays. Entorno de desarrollo. XCode 5.0 Creación y navegación de proyectos iOS en XCode. Uso eficiente de XCode. Análisis de rendimiento con Instruments. Establecer breakpoints e investigar problemas.

### **FORMACIÓN ESPECÍFICA**

#### **COCOA TOUCH. LIBRERÍAS DE CLASES**

Uso de las principales clases del framework Cocoa Touch. Beneficios del patrón de diseño del modelo vista-controlador (MVC). Definir delegados. Notificaciones. Uso de sintaxis de bloque para clarificar el código.

#### **VISTAS Y CONTROLADORES DE VISTA. INTERFAZ GRÁFICA DE USUARIO**

Diseño de vistas con Storyboards y Segues en el Interfaz Builder. Conectar vistas y controladores de vista. Navegación entre escenas en código y con el Interfaz Builder. Definir acciones para responder a eventos de control. Uso de autolayout para crear interfaces robustas y flexibles. Audiencia de destino internacional con localización. Añadir imágenes, gestos y animaciones a la app. TableViews: gestión de filas de datos tabulares. Crear una fuente de datos para llenar una tabla. Insertar y eliminar filas de tabla. Organizar datos en secciones de tabla. Controladores de navegación y barra de pestañas. Controladores de contenedores. Uso de controladores de navegación para crear apps jerárquicas. Uso de controladores de pestañas para crear apps multi-modo. Teclado virtual y pantalla táctil. Configurar teclado. Crear un delegado para gestionar entradas de teclado. Detectar y responder a toques, deslizamientos y otros gestos.

## **SERVICIOS WEB REMOTOS**

Hacer solicitudes asíncronas a una interfaz RESTful. Usar los datos JSON de la respuesta.

## **CORE DATA. MODELADO DE DATOS Y PERSISTENCIA**

Herramientas de modelado para apps de datos persistentes. Crear, leer (fetch), actualizar y eliminar entidades persistentes. Gestión de relaciones de modelo.

## **TRANSICIÓN A IPAD**

Uso del controlador de vista dividida y popovers específicos de apps iPad. Transformar una app para iPhone en una app Universal que se pueda ejecutar en ambas plataformas.

## **RENDIMIENTO Y DEBUGGING**

Evitar y gestionar advertencias de memoria. Usar Instruments para solucionar problemas de memoria y perfilar el código. Usar herramientas de depuración en XCode. Solucionar errores y advertencias comunes.