



# TEMARIO DE JAVA INTERMEDIO

PROGRAMA DE TECNOLOGÍA EN CÓMPUTO

JUNIO 2017

---

## 1. Excepciones

- a. Errores y Excepciones
- b. Excepciones básicas
- c. Capturar una excepción: palabras reservadas try, catch y finally
- d. Lanzar y especificar excepciones: palabras reservadas throw y throws
- e. Crear Excepciones Propias

## 2. Sistema de E/S de Java: java.io

- a. Concepto de flujo o stream en Java
- b. Tipos de flujo
- c. Flujos de bytes (Byte Streams): Clases abstractas InputStream y OutputStream
- d. Flujos de caracteres (Character Stream): Clases abstractas Reader y Writer
- e. Leer la entrada estándar de datos
- f. Obtener un flujo de caracteres a partir de uno de bytes: Clase InputStreamReader
- g. Buffer para caracteres: Clase BufferedReader
- h. Manejo de archivos y directorios: Clase File
- i. Escribir y leer datos de archivos: Clases FileOutputStream y FileInputStream
- j. Otros ejemplos básicos de aplicaciones de flujos
- k. Copia de archivos a nivel de bytes

## 3. Colecciones de objetos: Java Framework Collection

- a. Introducción a las colecciones y estructuras de datos
- b. Manejo de arreglos mediante la clase java.util.Arrays
- c. Interfaces base del JFC y sus métodos
  - i. Interfaz List
  - ii. Interfaz Map
  - iii. InterfazSet





# TEMARIO DE JAVA INTERMEDIO

PROGRAMA DE TECNOLOGÍA EN CÓMPUTO

JUNIO 2017

- d. Clases de implementación de las interfaces base.
  - i. Clases ArrayList, Vector y Stack
  - ii. Clase HashMap y Hashtable
  - iii. Clase HashSet

#### 4. Programación multihilo: Threads

- a. ¿Qué es la programación multihilo?
  - i. Threads en java
  - ii. Diagrama de estados de los threads
  - iii. Herencia de la clase Thread y manipulación de hilos
  - iv. Métodos run() y start()
- c. El productor consumidor:
  - i. Sincronización de métodos
- d. Interfaz Runnable

#### 5. Computación Distribuida: java.net

- a. Conceptos básicos de la comunicación con Sockets TCP/IP
- b. Direcciones IP y puertos
- c. ¿Qué es un socket?
- d. Clase Socket y SocketServer
- e. Leer y escribir en los sockets: sockets y streams
- f. Servir a múltiples clientes: sockets y threads

#### 6. RoboCode, aplicando Java a batallas con robots

- a. ¿Qué es RoboCode?
- b. API
- c. Funcionalidad
- d. Torneo de RoboCode

