

SIMULACIONES

IV + IoT + DL

Industria 5.0

Copyright 2026 para:

Diego L. Aristizábal Ramírez

Profesor, Facultad de Ciencias, Departamento de Física

Universidad Nacional de Colombia

Medellín

INTRODUCCIÓN

Bases de Industria 5.0

1

Temas

- ✓ INTRODUCCIÓN
- ✓ INDUSTRIA 5.0
- ✓ ¿QUÉ ES LA INSTRUMENTACIÓN VIRTUAL (IV)
- ✓ ¿QUÉ ES ANDROID?
- ✓ ¿CUÁL ES EL LENGUAJE PARA PROGRAMAR EN ANDROID?
- ✓ PROGRAMACIÓN ORIENTADA A OBJETOS
- ✓ PATRONES DE DISEÑO
- ✓ IDE: ANDROID STUDIO
- ✓ ¿QUÉ ES EL IoT?
- ✓ ¿QUÉ ES CLOUD COMPUTING?
- ✓ ¿QUÉ ES EDGE COMPUTING?
- ✓ ¿QUÉ ES EL DEEP LEARNING?
- ✓ ¿QUÉ SON LOS SMARTWATCHES Y WEAR OS?
- ✓ CONOCIMIENTO PREVIO
- ✓ METODOLOGÍA DE ESTAS NOTAS
- ✓ EL CURSO

INTRODUCCIÓN

La formación de un **ingeniero físico** en el contexto actual exige comprender y aplicar las tecnologías que están transformando la industria y la investigación científica. La **Instrumentación Virtual (IV)** permite diseñar y controlar experimentos y sistemas de medición con gran flexibilidad, superando las limitaciones de la instrumentación física tradicional. El **Internet de las Cosas (IoT)**, por su parte, conecta sensores y dispositivos en redes distribuidas que generan datos en tiempo real, posibilitando el monitoreo y la automatización de procesos tanto en laboratorios como en entornos industriales.

Finalmente, el **Deep Learning (DL)** ofrece la capacidad de analizar esos datos de manera inteligente, extrayendo patrones ocultos, optimizando procesos y habilitando predicciones con alta precisión.

Estas tecnologías se complementan con el desarrollo de aplicaciones móviles en **Android**, donde los ingenieros físicos pueden implementar algoritmos, sistemas de control e interfaces gráficas que llevan la experimentación y la instrumentación más allá del laboratorio. Además, el estudio de **Cloud Computing** y **Edge Computing** permite entender cómo la infraestructura de datos soporta la escalabilidad y la inmediatez de las soluciones modernas. En este escenario, los dispositivos inteligentes como **smartwatches** y **Wear OS** representan ejemplos concretos de integración entre instrumentación, conectividad y aprendizaje automático.

De esta manera, el curso prepara a los futuros ingenieros físicos para enfrentar los desafíos de la **Industria 5.0**, donde el conocimiento profundo de la física se combina con herramientas digitales de vanguardia para crear soluciones innovadoras, sostenibles y con un alto impacto en la sociedad

INDUSTRIA 5.0

La **Instrumentación Virtual (IV)**, el **Internet de las Cosas (IoT)** y el **Deep Learning (DL)** representan hoy un eje estratégico en la transición hacia la **Industria 4.0**. La IV ofrece la posibilidad de crear instrumentos digitales programables y adaptativos, con los que es posible medir y controlar variables físicas de manera flexible y precisa. El IoT extiende estas capacidades al interconectar sensores y sistemas en tiempo real, permitiendo la adquisición y transmisión masiva de datos desde entornos físicos a plataformas digitales distribuidas. Por su parte, el DL añade la dimensión analítica, capaz de descubrir relaciones ocultas en los datos, optimizar procesos y generar predicciones con un nivel de autonomía creciente. En conjunto, estas tecnologías no solo modernizan la instrumentación y el control de procesos industriales, sino que preparan el terreno para la **Industria 5.0**, donde la interacción colaborativa entre humanos y máquinas, sumada a criterios de sostenibilidad, marcará el rumbo de la innovación.

¿QUÉ ES LA INSTRUMENTACIÓN VIRTUAL (IV)?

Es la instrumentación diseñada con una combinación de hardware (sensores y tarjetas de adquisición de datos) y software en donde procesos digitales de adecuación, filtraje, comunicaciones y visualización están esencialmente centradas en el software. Este modelo

de desarrollo fue implantado por la **National Instruments** en los 80's del siglo XX en su maravillosos software **LabVIEW** (Chacón, 2002).

Ventajas de esta tecnología son:

- ✓ Disminución de costos.
- ✓ Rápido desarrollo.
- ✓ Adaptabilidad de los instrumentos

Estas notas están orientadas a dar los elementos básicos del desarrollo de este tipo de tecnología usando los dispositivos móviles **ANDROID**.

En la Figura 1 se ilustra un sonómetro virtual el cual está comunicado con la tarjeta de sonido de un dispositivo móvil.



Figura 1

¿QUÉ ES ANDROID?

Es un Sistema Operativo (SO) diseñado por Google para dispositivos móviles y basado en el **SO Linux**. Sus equivalentes de otras compañías son el **SO APPLE IOS** y el **SO WINDOWS PHONE**. Datos del 2025 le dan el 72.0% del mercado a **ANDROID**.

¿CUÁL ES EL LENGUAJE PARA PROGRAMAR EN ANDROID?

El lenguaje Java es el que preferiblemente se usa para programar en **ANDROID** debido a que éste es el lenguaje nativo de este sistema operativo.

Aplicación web: Usa lenguajes web, habitualmente ***HTML5 + CSS + JavaScript***, para poder hacer tareas muy sencillas, como una conversión de un sitio online a la pantalla del teléfono. En su contra está que no son capaces de aprovechar todo el hardware del dispositivo, como la cámara.

Aplicación nativa: La más potente. Usa el lenguaje del sistema operativo de la plataforma, logrando aprovechar todo el potencial del dispositivo, tanto a nivel de hardware como de software. En el caso de Android, el lenguaje es **Java**.

Nota: Java es diferente a JavaScript

Java es el leguaje preferido para desarrollar las aplicaciones (Apps) para ANDROID. Es cierto que el Java que se usa para Android no es exactamente el mismo, pero se diferencia muy poco del Java original. Por lo que sí hay interés en desarrollar aplicaciones para el sistema operativo de Google, aprender Java es un paso que sí o sí, hay que dar.

Actualmente se ha liberado por fin el lenguaje **KOTLIN** como lenguaje oficial en Android al nivel de Java. Sin embargo, el autor de estas notas desarrollará el curso en Java. Kotlin aunque es un lenguaje poderoso para programar también los móviles android, aún es muy nuevo. De todas formas, los programas desarrollados en cualquiera de los dos lenguajes, Java o Kotlin, se pueden pasar fácilmente del uno al otro.

PROGRAMACIÓN ORIENTADA A OBJETOS

La **programación orientada a objetos** (**OOP** por sus siglas en inglés) es un paradigma de programación modelada de la misma manera que el mundo que denominamos real. Este mundo está compuesto de **objetos** los cuales a su vez están conformados por objetos más pequeños. Este paradigma permite la generalización a través del concepto de **clase**. Una clase es la generalización de las **propiedades** y **comportamientos** (métodos, funcionalidad) de un determinado grupo de objetos. Por ejemplo una clase denominada **Polea** encapsula propiedades de las poleas como color, radio, masa, fricción en sus ejes...

y comportamientos como rotación y traslación. Este paradigma de programación debido a su grado de abstracción permite rápidos desarrollos de aplicaciones muy modulares, robustas y versátiles a través de los conceptos de encapsulamiento, herencia y polimorfismo (Lemay L. y Perkins C., 1998).

El **C⁺⁺** (creado en 1979, Bjarne Stroustrup) fue uno de los primeros lenguajes de programación orientada a objetos. Otro es el espectacular lenguaje de programación **JAVA** (creado por James Gosling, 1995), propiedad inicial de Sun Microsystems y luego de Oracle, es multiplataforma y es software libre.

Por las razones expuestas en el párrafo anterior, en estas notas la programación en ANDROID se hará con base en el lenguaje Java.

5

PATRONES DE DISEÑO

Diseñar software orientado a objetos es “difícil”, y diseñar software orientado a objetos reutilizable es todavía “más difícil”. Los denominados **patrones de diseño** están orientados a que el software desarrollado sea escalable, reutilizable y sea fácil su actualización.

El patrón de diseño de software que se empleará para la elaboración de las simulaciones será el **MVC** (**M**odelo-**V**ista-**C**ontrolador). En éste las clases responsables del modelo físico-matemático (solución de ecuaciones diferenciales y cálculos en general) se agrupan en un paquete denominado **MODELO**, las clases responsables de las visualizaciones por pantalla (gráficas, objetos dibujables) se agrupan en un paquete denominado **VISTA** y las clases responsables del control (cambio de parámetros, activación y desactivación de la simulación...) se agrupan en un paquete denominado **CONTROL**.

Los instrumentos virtuales deben poderse comunicar con el hardware (sensores, tarjetas de adquisición, puertos de comunicación, redes...) y para esto el diseño del software debería emplear patrones de diseño tales como los denominados **CLIENTE-SERVIDOR**, **TUBERÍA-FILTRO**, **MAESTRO-ESCLAVO**, **PUBLICADOR-SUSCRIPTOR**. Éstos se eligen con base en la aplicación que se desarrollará. Con base en esto se agrupan las clases y se asignan tareas específicas a ellas que el protocolo de estos patrones exige.

IDE: ANDROID STUDIO

Android Studio es la herramienta de desarrollo (IDE) oficial de Google para crear aplicaciones (apps) en dispositivos móviles Android. Es rápida, completa, gratuita y se adapta tanto a principiantes como a desarrolladores avanzados. En este curso se trabajará con la versión **android-studio-2025.1.3.7** la cual se puede obtener del sitio oficial:

developer.android.com/studio

De no tener acceso allí a esta versión, pueden obtenerla a través de links suministrados en el classroom del curso

¿QUÉ ES EL IoT?

IoT (por sus siglas en inglés, **Internet of Things**) describe la red de objetos físicos (cosas) que están integrados con sensores, software y otras tecnologías con el fin de conectar e intercambiar datos con otros dispositivos y sistemas a través de Internet. Estos dispositivos van desde objetos domésticos comunes hasta herramientas industriales sofisticadas. Para el finales del 2025 se espera esten conectados vía **IoT** 31 mil millones de dispositivos.

Esta tecnología permite que mediante la unión de la informática, los sensores de muy bajo costo y baja potencia de consumo, la computación en la nube, el big data, la inteligencia artificial y las tecnologías móviles (por ejemplo, la tecnología móvil **5G** le dará un impulso exponencial al **IoT**), las cosas físicas pueden compartir y recopilar datos con una mínima intervención humana, permitiendo un análisis de estos para obtener información muy valiosa y con muy alta confiabilidad. Esto conforma la denominada cuarta ola de la revolución industrial o **Industria 4.0**.

¿QUÉ ES CLOUD COMPUTING?

Computación en la Nube o simplemente la **Nube** (del inglés **Cloud Computing**) es el uso de servidores remotos conectados a internet para almacenar, administrar y procesar datos. En lugar de depender de un servicio físico instalado, se tiene acceso a una estructura donde el software y el hardware están virtualmente integrados.

¿QUÉ ES EDGE COMPUTING?

Computación en el Borde (del inglés, **Edge Computing**) es un nuevo paradigma, en donde el procesamiento y almacenamiento de los datos que producen las aplicaciones ocurre lo más cerca posible al usuario. El término “**Edge**”, que significa borde, se refiere a llevar el cómputo al límite, de acercar sus beneficios a los usuarios. En otras palabras es como llevar la **Nube** lo más cerca del usuario. Esto disminuye la latencia (tiempo invertido en enviar un paquete a la red y luego recibir su respuesta) a niveles tan bajos como los necesarios para tomar decisiones en tiempo real (menores a 10 ms). También disminuye el ancho de banda empleado.

La **Computación en el Borde** es posible hoy por la llegada de la **tecnología 5G** trabajando en conjunto con las redes de **fibra óptica**.

¿QUÉ ES EL DEEP LEARNING?

El **Deep Learning (Aprendizaje Profundo)** es una rama de la inteligencia artificial, en particular del aprendizaje de máquina (**Machine Learning**), que busca imitar cómo funciona el cerebro humano mediante redes de “neuronas” artificiales conectadas entre sí. La clave está en que estas redes tienen muchas capas (de ahí el término “profundo”), lo que les permite aprender automáticamente patrones muy complejos en imágenes, sonidos o textos. Es precisamente en el **Deep Learning** donde se han logrado los mayores avances en inteligencia artificial en los últimos años, dando lugar a aplicaciones sorprendentes como los traductores automáticos, el reconocimiento facial en los celulares o los autos que se conducen solos.

En esta notas la **redes neuronales** se entrenarán en Python o se obtendrán ya entrenadas. Luego se usarán en aplicaciones Android.

¿QUÉ SON LOS SMARTWATCHES Y WEAR OS?

Un **smartwatch** o **reloj inteligente** es un reloj que posee funcionalidades que van mas allá de las de un reloj básico y que hacen de él un dispositivo móvil multifunción. Los smartwatch funciones se pueden sincronizar con celulares y tabletas, o bien de modo independiente. En ellos, además de poseer múltiples sensores, pueden ejecutar aplicaciones dándoles posibilidad de ser una herramienta tecnológica para ser usada, entre otros, en instrumentación virtual.

Wear Os es sistema operativo **Android** de Google para todos los dispositivos denominados **wearables**¹, entre ellos, los relojes inteligentes. Es compatible con comunicaciones Bluetooth, WiFi, NFC, LTE, etc.

¹ Puede que la palabra **wearable** resulte un poco ajena a vocabulario diario, pero la realidad es que se convive con ella cada vez más. Es un término que viene del idioma inglés y que significa, “vestible”. Por eso, cuando se habla de dispositivos **wearables** se refiere a los relojes, pulseras, auriculares, gafas, zapatillas, llaveros o cualquier otro accesorio o prenda de vestir que se llevan encima y que sea capaz de realizar alguna tarea adicional al “reporte de la hora” como ofrecer datos de interés, como estado de salud, rutinas, lugares, noticias, previsión meteorológica

CONOCIMIENTO PREVIO

Para abordar el curso al cual se refieren estas notas se supone que se tienen los siguientes conocimientos previos:

- ✓ **Elementos básicos de programación en algún lenguaje:** algoritmia computacional y sintaxis básica (tipo de datos –incluyendo rango de tipo de números-, épsilon de máquina, overflow, underflow, sentencias de flujo de programas –for, if else, while...–).
- ✓ **Deseables fundamentos en electrónica y fundamentos en Deep Learning.**

METODOLOGÍA DE ESTAS NOTAS

En estas notas se hace especial énfasis en “trucos” diseñados específicamente (por el autor de las notas) para facilitar los desarrollos de aplicaciones en los dispositivos móviles en donde una de las principales dificultades, sino la mayor, es la diversidad en las especificaciones técnicas de éstos como por ejemplo las pantallas de diferentes tamaños y resoluciones. Adicionalmente en las mismas se propone una jerarquización de paquetes, clases, métodos dentro de las clases que han permitido realizar desarrollos con mucha versatilidad (al autor).

Las notas se dividieron en módulos de enseñanza-aprendizaje. Estos fueron preparados con el fin de servir como uno de los materiales didácticos que se podrá emplear para facilitar el desarrollo del curso. Este material no tiene el formato de LIBRO, más bien son NOTAS DE CLASE que el estudiante complementará con las exposiciones del profesor y otros materiales que serán sugeridos en la medida que el curso avance.

El curso se desarrollará en sesiones de 4 horas: 1 hora de evaluación, 1 hora de exposición magistral y 2 horas de actividades que desarrollan los estudiantes. En cada sesión se pueden abordar varios módulos. Al final de cada módulo se presenta un taller para ser resuelto. Se recomienda a los estudiantes que terminada cada clase **INMEDIATAMENTE Y CON MUCHO JUICIO** estudie el módulo correspondiente antes de su próxima clase.

Nota: Los módulos están en prueba y pueden presentar errores. Se agradece a los estudiantes que en caso de encontrar éstos lo hagan saber para realizar las debidas correcciones.

A continuación, se hace la lista de los módulos:

Parte I: Introducción

- ✓ MÓDULO # 1. INTRODUCCIÓN
- ✓ MÓDULO # 2. MI PRIMERA APP CON ANDROID STUDIO
- ✓ MÓDULO # 3. ARITMÉTICA COMPUTACIONAL
- ✓ MÓDULO # 4. DIBUJANDO CON ANDROID

Parte II: Fundamentos de OOP

- ✓ MÓDULO # 5. OOP –ENCAPSULAMIENTO Y POLIMORFISMO-
- ✓ MÓDULO # 6. OOP –HERENCIA E INTERFACE-
- ✓ MÓDULO # 7. ESTRUCTURA DE UNA APP ANDROID
- ✓ MÓDULO # 8. LA GUI

Parte III: Simulaciones en Física

- ✓ MÓDULO # 9: GENERANDO UNA LIBRERÍA EN ANDROID
- ✓ MÓDULO # 10: SIMULACIONES EN FÍSICA

Parte IV: Instrumentación Virtual

- ✓ MÓDULO # 11. IV CON LOS SENSORES DEL MÓVIL ANDROID
- ✓ MÓDULO # 12. FLUJOS DE DATOS y JSON
- ✓ MÓDULO # 13. IV + BLUETOOTH + ANDROID
- ✓ MÓDULO # 14. IV + IoT + ANDROID
- ✓ MÓDULO # 15. IV + IoT + SMARTWATCHES + WEAR OS

Parte V: ESP32 e Instrumentación Visrtual con Android

- ✓ MÓDULO # 16: ESP32 E IV: COMUNICACIÓN BLUETOOTH
- ✓ MÓSULO # 17: ESP32: COMUNIACIÓN IoT

Parte V: Deep Learning

- ✓ MÓDULO # 18: DEEP LEARNING CON ANDROID

Nota: para desarrollar este módulo los y las estudiantes ya habrán realizado un curso en PyTorch sobre Deep Learning en forma de autoaprendizaje.

FIN