



# Presentación curricular

## Ciclo 2

## Ruta de aprendizaje 2

## MINTIC

Ciclo 2: Programación básica en Java

AÑO 2021



UNIVERSIDAD  
SERGIO ARBOLEDA





# Introducción

El presente documento se ha elaborado en el **marco del proyecto MisiónTIC 2022 convenio de asociación No. 776 del 2021** como insumo para el desarrollo de la ruta de aprendizaje No. 2. Este documento y la información contenida en él son desarrollos exclusivos para el proyecto y la implementación de la ruta de aprendizaje No.2 MisionTIC 2022.

## La Universidad Sergio Arboleda

La **UNIVERSIDAD SERGIO ARBOLEDA** es una Institución de educación superior de carácter privado, sin ánimo de lucro, con Personería Jurídica reconocida mediante las Resoluciones Nos. 16377 de octubre de 1984, emanadas del Ministerio de Educación Nacional; orientada a formar profesionales en los distintos campos de la cultura, con proyección hacia la comunidad nacional e internacional, con sede en las ciudades de Bogotá D.C, Santa Marta y Barranquilla, y a nivel internacional, el Centro Internacional de Excelencia Internacional ubicado en la ciudad de Madrid – España y en la ciudad de Miami – U.S.A.. Actualmente cuenta con Acreditación Institucional de Alta Calidad Multi-campus otorgada mediante Resolución 3659 del 5 de abril de 2019, oferta ciento treinta y tres (133) programas, más de cuarenta y un (41) programas de pregrado, cuatro (4) programas de tecnología, sesenta (60) programas de especialización, veintisiete (27) maestrías, un (1) doctorado, y más de cien (100) programas de educación continuada.

Durante más de 35 años la Universidad Sergio Arboleda ha participado activamente en procesos de investigación y consultoría tanto en entidades privadas como públicas, en el ámbito nacional e internacional generando en sus aliados, importantes impactos a nivel de cambio y transferencia de conocimientos a través de la educación.

Con base en los Estatutos de la Universidad Sergio Arboleda (aprobados por la Resolución 6423 del 5 de agosto de 2011 del Ministerio de Educación Nacional), dentro de los objetivos de la misma se encuentra: "(...) d. *Adelantar y fomentar programas de investigación y extensión en las áreas de su especialidad y propender por su efectivo aprovechamiento y divulgación (...)*". Adicionalmente, de conformidad con la naturaleza académica de la Universidad, para el cumplimiento de sus objetivos y el desarrollo de sus funciones, "*podrá crear, dirigir, o administrar en todos los niveles, modalidades y metodologías, todas las actividades académicas, docentes, investigativas y de extensión o servicio conducentes a la producción, desarrollo y transmisión del conocimiento y de la cultura en todos los campos*".

De acuerdo con la Ley 30 de 1992, "*La extensión comprende los programas de educación permanente, cursos, seminarios y demás programas destinados a la difusión de los conocimientos, al intercambio de experiencias, así como las actividades de servicio tendientes a procurar el bienestar general de la comunidad y la satisfacción de las necesidades de la sociedad*".





Así las cosas, la Extensión en la Universidad Sergio Arboleda busca, en conjunto con los otros pilares de la Institución (Docencia, Investigación e Innovación), aportar a la proyección social a través de servicios, tales como convenios de ciencia, tecnología e innovación.

Con base en lo anterior, la Universidad ha dirigido sus esfuerzos a complementarse –teniendo en cuenta las necesidades actuales del país–, en las áreas de investigación e innovación que aportan conocimiento efectivo a las diferentes regiones de Colombia; logrando obtener resultados que han contribuido con el mejoramiento de la sociedad.

En el recorrido de más de 35 años, la Universidad Sergio Arboleda, se ha caracterizado por un sólido compromiso con el quehacer de su misión, liderando un proyecto de formación integral que descansa en cuatro pilares fundamentales: Humanismo, Excelencia Académica, Internacionalización y Emprendimiento e innovación.

Desde las aulas se han venido forjando individuos con profundos conocimientos en las diversas ciencias del saber, con un bagaje cultural que les permita identificar sus raíces, velar por los valores ancestrales y ser cultores del arte y las letras; aptos para participar en procesos de desarrollo aquí y más allá de las fronteras, y preparados para crear, emprender e innovar, condiciones indispensables en los profesionales actuales.

Año tras año la Universidad se ha sometido a un riguroso proceso de actualización del proyecto educativo institucional, de manera que responda a las tendencias y necesidades del mundo contemporáneo. Así lo evidencia su historia, las acciones presentes y las proyecciones al futuro.

## Estrategias pedagógicas de la ruta de aprendizaje No.2 en coherencia con las herramientas tecnológicas

Las estrategias pedagógicas planteadas por el programa de formación de la ruta de aprendizaje No.2 se enmarcan en el quehacer de las acciones pedagógicas y didácticas, que facilitan la aprensión del conocimiento y el desarrollo de competencias digitales y de programación, de manera dinámica, innovadora y creativa integrando las herramientas tecnológicas y el desarrollo curricular a la formación de los estudiantes que hacen parte del programa académico.

El programa promueve diferentes estrategias en la planificación y desarrollo que permitan promover el cambio y la innovación en los métodos de enseñanza bajo criterios tales como:

- Trabajo colaborativo en ambientes multiculturales e Interdisciplinarios
- Aplicación de herramientas tecnológicas en pro de la mejora y el aprendizaje





Como soporte al desarrollo pedagógico se cuenta con las siguientes tecnológicas:

- **LMS (Learning Management System - sistema para la gestión del aprendizaje) + herramientas integradas:** plataforma en la que se implementará con el fin de hacer más robusta y eficiente:
  - ✓ Las comunicaciones
  - ✓ Las interacciones académicas, sincrónicas como asincrónica,
  - ✓ La publicación de los materiales y las Guías de Estudio,
  - ✓ La participación y evaluación en cada uno de los módulos de aprendizaje.
- **Herramientas anti-plagio:** Pueden ser usada con fines de **revisión o evaluación** de productos académicos.
- **Inspección semiautomática de código:** permite verificar las reglas de buena calidad que debe tener un programa orientado por objetos como encapsulamiento, desacoplamiento, presencia de código duplicado y una buena estructura de los algoritmos.
- **Herramientas para videoconferencias:** disponible para la interacción, en tiempo real, entre estudiantes y profesores permitiendo la utilización de aulas presenciales en un *esquema sincrónico* por comunicación con voz, video y chat.

Aula Virtual: El Aula Virtual contará con el apoyo de Mastertech, como LMS (*Learning Management System* - sistema para la gestión del aprendizaje).

## Ciclo 2. Programación básica en Java Justificación

El segundo ciclo de formación, programación básica en Java, tiene como objetivo presentar a los estudiantes el paradigma de programación orientada por objetos, los fundamentos de las bases de datos relacionales y el patrón arquitectónico MVC como estrategias para modelar, diseñar e implementar soluciones de software a problemas reales.

La programación orientada por objetos es un paradigma que permite expresar soluciones en un lenguaje más cercano al contexto del problema. Es una forma de razonamiento que facilita la abstracción de los conceptos y permite independizarlos y relacionarlos de una manera simple, siendo uno de los paradigmas de programación más utilizados en el campo de la ingeniería de software.

La programación orientada por objetos se complementa con la utilización de las bases de datos relacionales, las cuales proporcionan una forma intuitiva de modelar y manipular información representada en tablas que se relacionan entre sí. Este tipo de bases de datos permiten representar grandes volúmenes de información a través de una estructura lógica que facilita el acceso y la manipulación de los datos a través del lenguaje de consulta estructurado (SQL). Las bases de datos relacionales son usadas por organizaciones de todo





tipo y tamaño convirtiéndose en una herramienta poderosa para satisfacer la necesidad de administrar y controlar diferentes fuentes de información.

Finalmente, el patrón MVC nos permite definir una arquitectura para las aplicaciones que desacopla los componentes de la interfaz gráfica, la lógica de negocio y la capa de persistencia. El patrón MVC facilita la definición de las responsabilidades de los diferentes componentes, facilitando la escalabilidad y la mantenibilidad de la aplicación. El uso del patrón MVC tiene como propósito presentar los conceptos básicos de arquitectura de software y la importancia de estos conceptos en la etapa de diseño.

En este curso, los estudiantes resolverán diferentes retos de programación combinando el paradigma de programación orientada por objetos, el uso de bases de datos relacionales y el patrón MVC que les permitirán adquirir las competencias necesarias en el área de la ingeniería de software.

## Objetivo general

Desarrollar en los estudiantes habilidades en programación mediante estudio del paradigma de programación orientado por objetos, las bases de datos relacionales y la arquitectura MVC como parte de su formación como desarrolladores Full Stack.

## Competencias por desarrollar

1. Modela e implementa soluciones de software utilizando el paradigma de programación orientada por objetos.
2. Representa un diseño orientado por objetos mediante el lenguaje UML
3. Captura requisitos funcionales a través de diagrama de casos de uso e historias de usuario
4. Diseña y ejecuta un conjunto de pruebas unitarias
5. Diseña y manipula bases de datos relacionales a través del lenguaje SQL
6. Utiliza la arquitectura MVC para desacoplar la lógica de negocio y la interfaz gráfica
7. Utiliza librerías y Frameworks para el desarrollo de aplicaciones
- 8.

## Resultados esperados de aprendizaje

- 1.1. Implementa correctamente atributos, métodos y constructores en un programa orientado por objetos.
- 1.2. Aplica correctamente los conceptos de encapsulamiento, interfaz y herencia en un programa orientado por objetos
- 1.3. Utiliza colecciones y arreglos de datos en la solución de problemas





- 2.1. Utiliza diagramas de clases para representar el diseño de un programa orientado por objetos
- 2.2. Reconoce las relaciones de dependencia, asociación y composición entre clases
- 3.1. Identifica los diferentes componentes de un diagrama de casos de uso
- 3.2. Reconoce las relaciones <<include>> y <<extends>> en un diagrama de casos de uso
- 3.3. Identifica los componentes de una historia de usuario
- 3.4. Reconoce el papel de los criterios de aceptación en una historia de usuario
- 4.1. Diseña pruebas unitarias caracterizando la salida de una función
- 4.2. Implementa pruebas unitarias para que sean ejecutadas de forma automática
- 5.1. Modela un conjunto de datos a través de diagrama E/R
- 5.2. Convierte un modelo E/R a un modelo relacional
- 5.3. Utiliza el lenguaje SQL para realizar las consultas CRUD a una base de datos
- 5.4. Conecta una base de datos relacional a un programa utilizando JDBC
- 6.1. Reconoce las responsabilidades de cada componente en una arquitectura MVC
- 6.2. Implementa correctamente la comunicación entre los diferentes componentes de una arquitectura MVC
- 7.2 Utiliza Frameworks para la construcción de aplicaciones con arquitectura MVC

## Contenido programático

El contenido del curso está organizada en los siguientes módulos:

- **Módulo 1.** Introducción a Java: Sesiones 1 a la 5
- **Módulo 2.** Programación orientada por objetos: Sesiones 6 a la 10
- **Módulo 3.** Interfaz gráfica con Swing: Sesiones 11 a la 15
- **Módulo 4.** Bases de datos relacionales con MySQL: Sesiones 16 a la 20
- **Módulo 5.** Arquitectura MVC: Sesiones 21 a la

## Contenido programático ciclo 2: programación básica en Java

Módulo	Sesión	Retos	Tema	Contenido
<b>Módulo 1: Introducción a JAVA</b>	1		Introducción a java	Introducción a java, instalación Netbeans, Tipos de datos, flujos de entrada y salida.
	2		Condicionales y funciones	If If-Else, Switch y funciones
	3		Estructuras de repetición	For, While, Do-While
	4		Arreglos y cadenas de caracteres	Operaciones entre cadenas; declaración y manipulación de arreglos; ciclo Foreach
	5	1	Refuerzo semana 1	Ejercicios
<b>Módulo 2: Programación orientada por objetos</b>	6		Clases y objetos	Declaración de clases: Métodos y atributos; Instancias: creación y manipulación; Encapsulamiento
	7		Herencia y polimorfismo	Herencia Polimorfismo por mensaje y herencia; clases abstractas e Interfaces
	8		Colecciones y pruebas unitaria con JUnit	ArrayList, HashMap, Pruebas unitarias con JUnit
	9		Lenguaje UML	Diagrama casos de uso, Historias de usuario, Diagrama de clases; composición, agregación,
	10	2	Refuerzo semana 2	Ejercicios
<b>Módulo 3: Interfaz gráfica con Swing</b>	11		Diseño de interfaz de usuario (UI)	Introducción a Swing, Diseño de interfaces mediante Wireframes (balsamiq)
	12		Contenedores (Swing)	JFrame y JPanel; propiedades generales; Layout
	13		Componentes (Swing)	Botones, cuadros de diálogo, selección, campos, áreas de texto, etc.
	14		Manejo de eventos	Captura de eventos: Listeners
	15	3	Refuerzo semana 3	Ejercicios
<b>Módulo 4: Bases de datos Relacionales con MySQL</b>	16		Introducción a las bases de datos relacionales	Introducción BD relacionales. Bd Relacionales vs NoSQL. Instalación MySQL
	17		Modelo E/R y modelo relacional	Modelado: Diagrama E/R, Modelo Relacional
	18		Normalización	Formas normales 1FN, 2FN, 3FN, FNBC
	19		Introducción a SQL	CRUD consultas simples - SQL
	20	4	Refuerzo semana 4	Ejercicios
<b>Módulo 5: Arquitectura MVC</b>	21		Arquitectura MVC	Arquitectura de software;Patrón MVC; separación de responsabilidades
	22		JDBC	Introducción JDBC; Configuración JDBC; CRUD usando JDBC
	23		Spring Boot	Framework de desarrollo Spring Boot para Java
	24		MVC con Spring Data JDBC	Implementación de MVC con Spring-jdbc en Java
	25	5	Refuerzo semana 5	Ejercicios

# Metodología

**Clase Magistral:** el docente hará inducción a los temas, desarrollará ejemplos y ejercicios complementarios para los estudiantes sobre los temas expuestos.

**Recursos en plataforma:** el estudiante complementará la información vista en clase, orientándola hacia el desarrollo de ejercicios en Java disponibles en la plataforma *Mastertech* y material complementario como: videos, documentos y ejercicios resueltos.

**Desarrollo de tutorías:** el estudiante tendrá a su disposición tutores que resolverán las dudas en sesiones sincrónicas; estas tutorías se enfocan en el desarrollo de ejemplos y ejercicios que ayuden a afianzar los conceptos vistos en clases magistrales.

**Desarrollo de retos:** cada módulo estará acompañado por un reto diseñado para probar las habilidades adquiridas a lo largo del ciclo. Estos retos se desarrollarán de forma incremental de acuerdo a los contenidos vistos en cada módulo.

## Actividades de aprendizaje

50 horas de sesiones sincrónicas acompañadas por un formador específico, cada sesión de dos horas de duración. Las temáticas asociadas a cada una de las sesiones se presentaron en la sección contenido programático.

120 horas de trabajo independiente de programación entre las cuales podrán solicitar el acompañamiento de un tutor específico, realizar ejercicios propuestos extra clase y avanzar en el desarrollo de los retos pedagógicos propuestos.

## Evaluación

La evaluación del ciclo 2: programación básica en Java corresponde a la evaluación de los cinco (5) retos académicos y a la evaluación del curso de inglés. Las evaluaciones se hacen en la escala de 0.0 al 5.0, donde cero (0.0) es la mínima nota y cinco (5.0) la máxima nota, la nota mínima aprobatoria del ciclo es de tres (3.0).

La distribución del porcentaje de la nota final se hará de la siguiente manera:

Reto 1: 10%

Reto 2: 10%





Reto 3: 20%  
Reto 4: 20%  
Reto 5: 20%  
\*Curso de Inglés: 20%



## Recursos

### LMS Mastertech

Es una plataforma LMS (*Learning Management System*, sistema de gestión de aprendizaje) que **ofrece cursos con** contenidos actualizados, contruidos por expertos y certificados por la Universidad Sergio Arboleda en convenio con líderes de cada sector. La plataforma ofrece cursos 100% virtuales, de manera que el estudiante pueda ir a su ritmo y siempre contará con el apoyo de toda la comunidad o de expertos (Formadores y tutores) para apoyar el proceso y asegurar que los estudiantes logren sus objetivos de aprendizaje.

## Referencias Bibliográficas

### Head First Java, 2<sup>nd</sup> Edition

Sierra, K., & Bates, B. (2013). Head first java. O'Reilly Media, Inc.

### Clean code: a handbook of agile software craftsmanship.

Martin, R. C. (2009). Clean code: a handbook of agile software craftsmanship. Pearson Education.

### Software Engineering

Sommerville, I. (2016). Software Engineering GE. Pearson Australia Pty Limited.

### Bases de datos: diseño, implementación y administración

Coronel, C., Morris, S., & Rob, P. (2011). Bases de datos: diseño, implementación y administración. Cengage Learning Editores.

### UML@ classroom: An introduction to object-oriented modeling

Seidl, M., Scholz, M., Huemer, C., & Kappel, G. (2015). UML@ classroom: An introduction to object-oriented modeling. Springer.

### Plataforma MasterTech

Curso Java para todos. Enlace: <https://mastertech.usergioarboleda.edu.co/>

