



Fundamentos de Programación (PRY2201)

Plan de la asignatura (PA)

Índice

Bienvenida.....	3
Propósito	4
Ruta de aprendizaje.....	5
Evaluación	13
Roles del equipo	14
Glosario	15
Bibliografía.....	18

Bienvenida

¿Te has preguntado alguna vez cómo es posible que las aplicaciones en tu teléfono o computadora funcionen de la manera en que lo hacen? En la asignatura de Fundamentos de Programación, los estudiantes conocerán los pilares fundamentales que hacen posible esta maravilla digital.

A través de esta experiencia, aprenderán a elaborar estructuras de datos temporales en lenguaje de programación Java, utilizar diversas estrategias para la construcción de algoritmos, automatizar tareas, entre otras cosas.

Cuando terminen esta experiencia de aprendizaje, no solo habrán aprendido a programar, sino que también habrán desarrollado habilidades críticas de pensamiento lógico y resolución de problemas que le serán útiles en diversas áreas de su vida profesional.

Propósito

Esta asignatura está enmarcada en un modelo pedagógico basado en Resultados de Aprendizaje (RA) y se declara como propósito:

- Aplicar, a través del pensamiento lógico y la creación de algoritmos, los pilares fundamentales de la programación, basándose en la construcción de estructuras de datos temporales, arreglos unidimensionales o bidimensionales y listas para el almacenamiento de información.

Estructura por experiencia

Esta asignatura consta de 3 experiencias de aprendizaje, distribuidas en 8 semanas y la semana 9 para rendir la “Evaluación Final Transversal”.

Experiencia 1: Creando mi lógica aplicada

Durante la experiencia de aprendizaje 1 los y las estudiantes trabajarán en el diseño el primer algoritmo, modelando y resolviendo problemas algorítmicos con expresiones aritméticas y lógicas. Esta experiencia responde al resultado de aprendizaje 1: Utiliza estrategias de abstracción para la construcción de algoritmo, aplicando pseudocódigo con el objetivo de dar solución a problemáticas planteadas.

- **Semanas:** 1, 2 y 3

Experiencia 2: Explorando mi software de programación Java

Durante la experiencia de aprendizaje 2 los y las estudiantes revisarán los mecanismos para dirigir el flujo de ejecución de un programa, utilizando variables y entornos de desarrollo integrados. Esta experiencia responde al resultado de aprendizaje 2: Aplica elementos básicos de Java para el desarrollo de programas que solucionen problemáticas planteadas.

- **Semanas:** 4, 5 y 6

Experiencia 3: Programando arreglos, listas y funciones

Durante la experiencia de aprendizaje 3 los y las estudiantes gestionarán datos con listas en Java, optimizando la gestión de estos datos. Esta experiencia responde al resultado de aprendizaje 3: Construye estructuras de datos temporales, utilizando arreglos para el almacenamiento de información y el trabajo con problemáticas que requieren la mantención en datos de memoria.

- **Semanas:** 7 y 8

Considera: En cada una de estas semanas contarás con la Guía de Aprendizaje, la cual tendrá información relevante sobre las temáticas abordadas y te dará herramientas para que puedas rendir correctamente las actividades declaradas. Esta información podrás visualizarla igualmente de manera interactiva con los recursos de aprendizajes.

Ruta de aprendizaje

Este espacio contiene toda la información que necesitarás para conocer la asignatura, y así organizar el tiempo que requieras para adquirir los aprendizajes dispuestos en las experiencias propuestas.

Experiencia 1: Creando mi lógica aplicada

	Resultado de Aprendizaje (RA)	Indicador de Logro (IL)	Actividad / Evaluación	Recursos
Semana 1: Mi primer algoritmo	RA1. Utiliza estrategias de abstracción para la construcción de algoritmo, aplicando pseudocódigo, con el objetivo de dar solución a problemáticas planteadas.	IL1. Identifica estrategias de abstracción y tipos de algoritmos aplicables en la construcción de algoritmos.	Encargo: Creando mi primer algoritmo <ul style="list-style-type: none">• Individual• Formativa	<ul style="list-style-type: none">• Mi primer algoritmo.• Capítulo 1: All about Java / Capítulo 2: All about Software.• Capítulo 1: Introducción a Java.

Semana 2: Diseño y modelado de Algoritmos	RA1. Utiliza estrategias de abstracción para la construcción de algoritmo, aplicando pseudocódigo, con el objetivo de dar solución a problemáticas planteadas.	IL1. Identifica estrategias de abstracción y tipos de algoritmos aplicables en la construcción de algoritmos. IL2. Aplica pseudocódigo en la implementación de algoritmos, brindando una solución a las problemáticas planteadas.	Encargo: Diseñando y modelando Algoritmos <ul style="list-style-type: none"> • Individual • Formativa 	<ul style="list-style-type: none"> • Diseño y modelado de algoritmos • Ejemplo de programación estructurada • Capítulo 4: Making the most variables and their values • Capítulo 1: Introducción a Java
Semana 3: Problemas Algorítmicos con Expresiones aritméticas y lógicas	RA1. Utiliza estrategias de abstracción para la construcción de algoritmo, aplicando pseudocódigo, con el objetivo de dar solución a	IL1. Identifica estrategias de abstracción y tipos de algoritmos aplicables en la construcción de algoritmos.	Encargo: Resolviendo Problemas Algorítmicos <ul style="list-style-type: none"> • Individual • Sumativa 30% 	<ul style="list-style-type: none"> • Expresiones aritméticas y Expresiones lógicas • Mi primer programa en NetBeans • Capítulo 5: Controlling Program Flow with Decision-Making

	<p>problemáticas planteadas.</p> <p>IL2. Aplica Pseudocódigo en la implementación de algoritmos, brindando una solución a las problemáticas planteadas.</p> <p>IL3. Utiliza Variables, Expresiones Aritméticas y Lógicas en la Resolución de Problemas Algorítmicos, garantizando la exactitud y coherencia de sus soluciones.</p>		
--	--	--	--

Experiencia 2: Explorando mi Software de programación Java

	Resultado de Aprendizaje (RA)	Indicador de Logro (IL)	Actividad / Evaluación	Recursos
Semana 4: Mecanismos para dirigir el flujo de ejecución de un programa y toma de decisiones	RA2. Aplica elementos básicos de Java para el desarrollo de programas que solucionen problemáticas planteadas.	IL4. Utiliza Algoritmos en Java en la creación de programas funcionales que solucionen las problemáticas planteadas.	Encargo: Utilizando estructuras de control y condicionales en Java <ul style="list-style-type: none"> • Individual • Formativa 	<ul style="list-style-type: none"> • Estructuras de control y condicionales • GitHub • Chapter 6: Controlling Program Flow with loops. • Capítulo 1: Introducción a Java – 1.16 Sentencias de control: if.
Semana 5: Variables en Java	RA2. Aplica elementos básicos de Java para el desarrollo de programas que solucionen problemáticas planteadas.	L5. Emplea Variables en Java en la resolución de problemáticas planteadas, asegurando el correcto almacenamiento y manipulación de datos.	Encargo: Aplicando variables en Java <ul style="list-style-type: none"> • Individual • Formativa 	<ul style="list-style-type: none"> • Aplicaciones en java a partir de variables asociadas a datos. • Capítulo 1: Introducción a Java – Variables. • Capítulo 4: making the Most of Variables and Their Values.

<p>Semana 6: Entornos de Desarrollo Integrados (IDEs)</p>	<p>RA2. Aplica elementos básicos de Java para el desarrollo de programas que solucionen problemáticas planteadas.</p>	<p>IL4. Utiliza Algoritmos en Java en la creación de programas funcionales que solucionen las problemáticas planteadas.</p> <p>IL5. Emplea Variables en Java en la resolución de problemáticas planteadas, asegurando el correcto almacenamiento y manipulación de datos.</p> <p>IL6. Utiliza Entornos de Desarrollo (IDEs) en la escritura, depuración y aplicación de expresiones</p>	<p>Encargo: Depurando mi proyecto en el Entorno de Desarrollo Integrado (IDE)</p> <ul style="list-style-type: none"> • Individual • Sumativa 35% 	<ul style="list-style-type: none"> • Eficiencia de Entornos de Desarrollo Integrado (IDE).
--	--	--	--	---

		aritméticas en programas Java que solucionen problemáticas planteadas.		
--	--	--	--	--

Experiencia 3: Programando arreglos, listas y funciones

	Resultado de Aprendizaje (RA)	Indicador de Logro (IL)	Actividad / Evaluación	Recursos
Semana 7: Gestión de datos con Listas en Java	RA3. Construye estructuras de datos temporales, utilizando arreglos para el almacenamiento de información y el trabajo con problemáticas que requieren la mantención en datos de memoria.	IL7. Define arreglos en Java a través de listas, brindando la capacidad de acceder, agregar y eliminar elementos con la sintaxis e indexación adecuadas.	Encargo: Gestionando Datos con Listas en Java <ul style="list-style-type: none"> • Individual • Formativa 	<ul style="list-style-type: none"> • Gestión de datos con listas en Java • Chapter 11: Using Arrays to juggle values
Semana 8: Optimización de la gestión de datos en Java	RA3. Construye estructuras de datos temporales, utilizando arreglos para el almacenamiento de información y el trabajo con problemáticas que requieren la	IL8. Modifica arreglos en Java que permitan agregar, eliminar y actualizar elementos, garantizando la integridad y coherencia de los datos.	Encargo: Optimizando la Gestión de Datos en Java <ul style="list-style-type: none"> • Individual • Sumativa 35% 	<ul style="list-style-type: none"> • Optimizando listas y arreglos en Java • Optimizando el rendimiento en el manejo de datos primitivos • Capítulo 4: Programación por contrato, pruebas unitarias y diseño de algoritmos

	mantenimiento en datos de memoria.	IL9. Optimiza el rendimiento en el manejo de datos primitivos en Java a través de la selección de estructura de datos adecuada (arreglos o listas), en función de los requisitos de las problemáticas planteadas.		
--	------------------------------------	--	--	--

Evaluación

En cada una de las experiencias hay actividades formativas y al finalizarlas una actividad sumativa. A continuación, se detalla la ponderación de cada una de ellas:

Experiencia de aprendizaje	Nº1: Creando mi lógica aplicada
Semanas	1, 2 y 3
Tipos de evaluaciones	2 formativas 1 sumativa
Ponderación	30 % de las Experiencias

Experiencia de aprendizaje	Nº2: Explorando mi software de programación Java
Semanas	4, 5 y 6
Tipos de evaluaciones	2 formativas 1 sumativa
Ponderación	35 % de las Experiencias

Experiencia de aprendizaje	Nº3: Programando arreglos, listas y funciones
Semanas	7 y 8
Tipos de evaluaciones	1 formativa 1 sumativa
Ponderación	35 % de las Experiencias

Cada una de las experiencias totalizan un 100 %, sin embargo, se convierten en el 60% de la puntuación de la asignatura.

Finalmente tendrás que rendir la siguiente Evaluación final transversal para culminar la asignatura.

Semana	9
Situación evaluativa	Entrega de encargo
Tipo de evidencia	Producto
Ponderación	40 % de la asignatura

Roles del equipo

Docente: es el/la facilitador(a) y guía de los procesos de aprendizaje, colaborativos e individuales, de la asignatura. Como experto en la disciplina, uno de sus roles principales es guiar a los estudiantes en base a los materiales de aprendizaje dispuestos en el AVA, fomentando el avance del proceso formativo y entregando retroalimentación oportuna del desempeño del estudiante.

Tutor: La función de el/la tutor(a) online es acompañar y orientar al estudiante en aspectos técnicos, administrativos y también, en aquellos que pudieran estar interfiriendo en su rendimiento, a lo largo de su proceso formativo, por medio de los canales de comunicación online, dispuestos para ello.

Glosario

Estos conceptos pueden ser los más importantes que se declaran en la guía de aprendizaje y deben tener su fuente y ordenados alfabéticamente.

1. **Algoritmo:** secuencia finita de instrucciones bien definidas para realizar una tarea o resolver un problema.
2. **Algoritmos de búsqueda:** conjunto de técnicas para encontrar datos o elementos dentro de una estructura.
3. **Algoritmos de ordenamiento:** métodos para organizar los elementos de una lista o arreglo en un orden específico.
4. **Algoritmos estructurados:** técnicas de programación que facilitan la lectura, desarrollo y mantenimiento del código mediante el uso de bloques o estructuras claras.
5. **Arreglos en Java:** estructuras que permiten almacenar múltiples valores del mismo tipo en una única variable.
6. **Call Stack:** estructura de datos que almacena información sobre las funciones activas en un programa, incluyendo su orden de ejecución.
7. **Comentarios:** texto en el código fuente que explica el funcionamiento o intención de partes del código, no ejecutado por el compilador o intérprete.
8. **Concatenación de cadenas:** operación de unir dos o más cadenas de texto en una sola.
9. **Constantes:** variables cuyo valor no puede modificarse una vez que han sido inicializadas.
10. **Control de flujo:** instrucciones que determinan el orden en que se ejecutan las operaciones en un programa.
11. **Convenciones:** conjunto de reglas o recomendaciones para escribir código de manera que sea fácil de leer y mantener.
12. **Debugging:** proceso de identificar y corregir errores en el código fuente de un programa.

- 13. Declaración de una variable:** acto de definir una variable especificando su tipo y nombre, reservando así espacio en memoria.
- 14. Declaración y creación de listas y arreglos:** proceso de definir y asignar espacio en memoria para listas y arreglos, estructuras que almacenan múltiples elementos.
- 15. Eclipse:** entorno de Desarrollo Integrado (IDE) ampliamente usado para la programación en Java, facilitando la escritura, depuración y compilación de código.
- 16. Eficiencia de algoritmos:** medida de los recursos que un algoritmo necesita para ejecutarse, generalmente tiempo de ejecución y espacio de memoria.
- 17. Eficiencia y complejidad:** términos relacionados con la optimización del rendimiento de los algoritmos, evaluando su tiempo de ejecución y uso de memoria.
- 18. Entorno de Desarrollo Integrado (IDE):** aplicación que proporciona servicios integrales para facilitar la programación, como editor de texto, compilador, depurador, entre otros.
- 19. Entrada de datos:** proceso de recibir datos del exterior (usuario, archivo, red) para usarlos en un programa.
- 20. Estructura secuencial:** modelo de programación donde las instrucciones se ejecutan una después de otra en un orden lineal.
- 21. Estructuras condicionales:** instrucciones que permiten tomar decisiones en un programa, ejecutando diferentes bloques de código según se cumplan ciertas condiciones.
- 22. Estructuras de control:** construcciones del lenguaje de programación que dirigen el flujo de ejecución del programa (bucles, condicionales, etc.).
- 23. Estructuras de control en Java:** instrucciones como if, switch, for, while, que controlan el flujo de ejecución en programas Java.
- 24. Estructuras de datos temporales:** estructuras de datos usadas de forma transitoria dentro de un programa, generalmente con duración limitada al alcance de un bloque de código.
- 25. Estructuras iterativas:** estructuras de control que repiten un bloque de código múltiples veces, como los bucles for, while en Java.
- 26. Expresiones aritméticas:** combinaciones de números, operadores y paréntesis que representan una cantidad numérica.

- 27. Gestión de memoria:** prácticas y técnicas para asignar, usar y liberar memoria en la ejecución de programas.
- 28. Inicialización de una variable:** asignación de un valor inicial a una variable en el momento de su declaración.
- 29. Integridad y coherencia de datos:** asegurar que los datos sean precisos y consistentes a lo largo de todo su ciclo de vida.
- 30. Iteración a través de listas y arreglos:** recorrer elementos de listas o arreglos secuencialmente, accediendo a cada elemento.
- 31. Listas en Java:** estructuras de datos dinámicas que permiten almacenar elementos de forma ordenada, con capacidad de cambiar su tamaño.
- 32. Operaciones con arreglos:** acciones realizadas sobre arreglos, como la inserción, eliminación, o búsqueda de elementos.
- 33. Operadores:** símbolos que indican al compilador que realice operaciones matemáticas, lógicas o de otro tipo.
- 34. Operadores aritméticos:** símbolos utilizados en expresiones matemáticas para realizar operaciones como suma, resta, multiplicación, etc.
- 35. Operadores lógicos:** símbolos que se usan para formar expresiones lógicas, comúnmente para control de flujo, como AND, OR, NOT.
- 36. Optimización del código:** proceso de modificar un código para mejorar su eficiencia, en términos de velocidad de ejecución, uso de memoria, entre otros.
- 37. Prácticas de codificación:** conjunto de técnicas y recomendaciones que buscan mejorar la claridad, eficiencia y mantenibilidad del código.
- 38. Prácticas de codificación eficiente:** técnicas y estrategias para escribir código que se ejecuta de manera rápida y eficiente, utilizando los recursos de manera óptima.
- 39. Pseudocódigo:** representación de algoritmos utilizando una mezcla de lenguaje natural y convenciones de programación, facilitando su comprensión y diseño.
- 40. Rendimiento en el manejo de datos primitivos:** eficiencia en la manipulación de tipos de datos básicos como enteros, flotantes, booleanos, etc.
- 41. Resolución de problemas:** proceso de identificar, analizar y solucionar problemas utilizando métodos sistemáticos y lógicos.
- 42. Salida de datos:** proceso de enviar datos desde el programa hacia afuera, ya sea a la pantalla, a un archivo, a la red, etc.

- 43. Sintaxis de Java:** conjunto de reglas y estructuras que definen la estructura y la forma en que se escriben los programas en Java.
- 44. Tipos de datos:** diversas categorías en las que se pueden clasificar los datos en la programación, como enteros, decimales, booleanos, etc.
- 45. Tipos de datos básicos:** categorías elementales de datos que son intrínsecas a un lenguaje de programación, como int, float, char en Java.
- 46. Validación de resultados:** proceso de verificar que los resultados de un programa son correctos y cumplen con los requisitos o expectativas.
- 47. Variables:** espacios en la memoria que pueden contener datos que pueden cambiar durante la ejecución de un programa.
- 48. Ventana de variables:** en un IDE, es una herramienta que muestra las variables y sus valores mientras se está depurando un programa.
- 49. Selección de estructuras de datos:** proceso de elegir la estructura de datos más adecuada para resolver un problema específico, basándose en factores como la eficiencia y la complejidad.

Bibliografía

- Burd, B. (2022). Java for Dummies. New Jersey: John Wiley & Sons.
[https://bibliotecabuscador.duoc.cl/client/es_CL/default/search/detailnonmodal/ent:\\$002f\\$002fSD_ILS\\$002f0\\$002fSD_ILS:50163/ada?qu=java&qf=PUBDATE%09Fecha+de+publicaci%C3%B3n%092019-2023%092019-2023&d=ent%3A%2F%2FSD_ILS%2F0%2FSD_ILS%3A50163%7EILS%7E0&h=8](https://bibliotecabuscador.duoc.cl/client/es_CL/default/search/detailnonmodal/ent:$002f$002fSD_ILS$002f0$002fSD_ILS:50163/ada?qu=java&qf=PUBDATE%09Fecha+de+publicaci%C3%B3n%092019-2023%092019-2023&d=ent%3A%2F%2FSD_ILS%2F0%2FSD_ILS%3A50163%7EILS%7E0&h=8)
- Vegas Gertrudix, J. M. (2022). Java 17: Fundamentos prácticos de programación. Bogotá: Ediciones de la U.
[https://bibliotecabuscador.duoc.cl/client/es_CL/default/search/detailnonmodal/ent:\\$002f\\$002fSD_ILS\\$002f0\\$002fSD_ILS:50229/ada?qu=java&qf=PUBDATE%09Fecha+de+publicaci%C3%B3n%092019-2023%092019-2023&d=ent%3A%2F%2FSD_ILS%2F0%2FSD_ILS%3A50229%7EILS%7E1&h=8](https://bibliotecabuscador.duoc.cl/client/es_CL/default/search/detailnonmodal/ent:$002f$002fSD_ILS$002f0$002fSD_ILS:50229/ada?qu=java&qf=PUBDATE%09Fecha+de+publicaci%C3%B3n%092019-2023%092019-2023&d=ent%3A%2F%2FSD_ILS%2F0%2FSD_ILS%3A50229%7EILS%7E1&h=8)

Importante

Para visualizar la ubicación de tu asignatura, en conjunto a los prerrequisitos, pincha aquí para entrar a la malla de tu carrera:

<https://www.duoc.cl/duoconline/>



Reservados todos los derechos Fundación Instituto Profesional Duoc UC. No se permite copiar, reproducir, reeditar, descargar, publicar, emitir, difundir, de forma total o parcial la presente obra, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros) sin autorización previa y por escrito de Fundación Instituto Profesional Duoc UC
La infracción de dichos derechos puede constituir un delito contra la propiedad intelectual.