

Facultad	Facultad de Ingeniería		
Proyecto	CONVENIO DE ASOCIACIÓN No. 777 DE 2021 SUSCRITO ENTRE EL FONDO ÚNICO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES Y LA UNIVERSIDAD EL BOSQUE		
Nombre del Ciclo	PROGRAMACIÓN BÁSICA		
Periodo Académico	20211		
Ciclo Pre-requisito	FUNDAMENTOS DE PROGRAMACIÓN		
Ciclo Co-requisito			
Horas acompañamiento formador:	50	Horas Trabajo Independiente:	120
Docente			

EQUIPO DOCENTE

	NOMBRE	CORREO ELECTRÓNICO	HORARIO ATENCIÓN A ESTUDIANTES (Día – Hora)	LUGAR DE ATENCIÓN A ESTUDIANTES
Diseñado por	Helio Ramírez	hramireza@unbosque.edu.co		
Docentes				

Información Académica

1. Justificación

Programación Básica es un ciclo de formación en programación de computadores. En el ciclo los estudiantes fortalecen las habilidades de programar usando el paradigma de orientación a objetos. Durante la construcción de los programas, los estudiantes identifican, interpretan y señalan posibles transformaciones a las situaciones planteadas de su entorno inmediato.

El ciclo Programación Básica hace parte del proyecto de formación “Misión TIC 2022”. El proyecto de formación “Misión TIC 2022” se compone de una serie de ciclos a lo largo de ocho (8) meses, esta formación prepara al estudiante para el diseño y construcción de software y bases de datos integradas a una infraestructura tecnológica.

Programación Básica es el segundo ciclo de formación, en el se fortalecen los conceptos fundamentales de la programación (objeto, clase, relaciones, instrucciones de control, variables, tipos de datos, herencia, polimorfismo, excepciones, entre otros) y se construyen los programas utilizando el lenguaje JAVA, basados en el patrón de diseño MVC (modelo vista controlador) e introduciendo patrones adicionales como DAO, DTO y GRASP.

2. Perfil del Egresado

El ciclo de Programación Básica contribuye en el beneficiario en el desarrollo de las siguientes capacidades:

- Desarrollar de manera individual un programa orientado por objetos en Java, monousuario, para resolver los requerimientos planteados por un tercero, que use una base de datos relacional, con una arquitectura MVC y con una interfaz gráfica.
- Diseñar y utilizar una base de datos relacional simple usando SQL, y conectarla desde un programa usando JDBC.
- Diseñar una aplicación utilizando objetos, expresando dicho diseño con un diagrama de clases de UML, con diagramas de casos de uso, y con un esquema básico de historias de usuario.
- Diseñar y ejecutar un conjunto de pruebas unitarias para un programa.
- Entender y utilizar los conceptos de interfaz y de herencia como mecanismos de reutilización.
- Utilizar una arquitectura MVC para desacoplar la lógica de negocio de la interfaz de usuario.
- Utilizar librerías y frameworks que apoyen el desarrollo de software

3. Contenidos Generales

- Identificación de requerimientos y diseño de soluciones: Standalone
- Análisis y Diseño de soluciones: Standalone – Diagramas UML.
- Introducción al paradigma de programación orientada a objetos (POO)
- Clases, objetos, atributos, asociaciones y métodos.
- Patrón de Diseño MVC: Aproximación UEB
- Abstracción, Herencia, Polimorfismo, Sobrecarga de métodos, Sobreescritura de métodos Clases y Métodos Abstractos e Interfaces
- Diseño y Manejo de excepciones: Try-catch.
- Documentación de Código y Asignación de Responsabilidades: GRASP (Alta cohesión y Bajo acoplamiento).
- Interfaz Gráfica de Usuario (GUI) - Componentes de Interacción: Lienzo, Menús, Combo-box, Diálogos, entre otros.
- Persistencia: Archivos de Texto Plano, Archivos de Propiedades, Archivos Binarios.
- Persistencia: Bases de Datos, Conexión entre aplicaciones y bases de datos: JDBC
- Proceso de pruebas de la solución (Pruebas Unitarias e Invariantes) y Depuración de Programas.

4. Objetivos de aprendizaje

Dimensión de aprendizaje significativo	Objetivos de aprendizaje Los estudiantes
Conocimiento fundamental	Identificar los principios del paradigma de la programación orientada a objetos, tales como: objeto, clase, atributo, método, asociación relacionándolos con un lenguaje de programación basado en el mismo paradigma para construir diagramas de clases basados en el lenguaje UML como solución a un caso de estudio.

Aplicación	Emplear palabras reservadas de un lenguaje de programación orientado a objetos para construir programas con base en modelos diseñados previamente.
Integración	Usar experiencias propias y desarrollar nuevas habilidades y criterios a partir de la interacción con sus compañeros y los demás cursos.
Dimensión humana	Desarrollar actividades de manera grupal buscando con ello el aprendizaje por medio de la experiencia y conocimiento de los demás compañeros.
Compromiso	Reconocer la importancia de la aplicación efectiva de los conceptos vistos en clase en la resolución de problemas de programación.
Aprender a aprender	Investigar sobre nuevas tecnologías y casos de estudio empresariales que les permita comprender que deben mantenerse actualizados para poder adaptarse a los continuos cambios generados por la vertiginosa evolución de la tecnología.

5. Actividades generales de aprendizaje

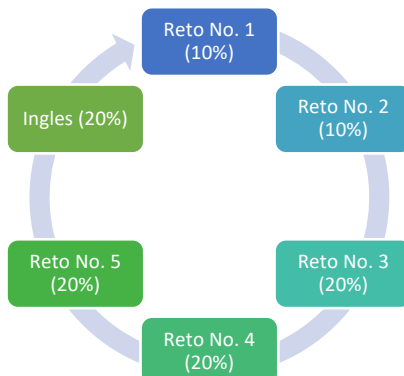
Cada una de las temáticas a trabajar durante el desarrollo del ciclo, se realiza con base en la propuesta de uno o varios casos de estudio definido por el docente. Adicionalmente se propone una serie de retos en el ciclo que deberá ser realizado por parte de los estudiantes acorde a las explicaciones y ejercicios prácticos dados por el docente.

1. Sesiones síncronas: el profesor con ayuda de recursos TIC expone ante los estudiantes la parte teórica del ciclo de formación. Los beneficiarios de forma individual deben aplicar los conocimientos teóricos para resolver los ejercicios que acompañan dichas sesiones.
2. Aprendizaje basado en resolución de casos: El docente proporciona a los estudiantes diversos enunciados en los cuales se describe la práctica a realizar a nivel grupal o individual. Cada una de las actividades prácticas a realizar está centrada con la temática de la asignatura y debe ser entregada su solución de manera parcial en fechas establecidas previamente.
3. Sesiones de tutoría: Los beneficiarios podrán solicitar al docente un espacio en un horario determinado para clarificar las dudas e inconvenientes que ocurran en el desarrollo de la asignatura o con la solución de los ejercicios propuestos. Durante este tipo de sesiones, se llevará a cabo el seguimiento y supervisión del proceso.
4. Búsqueda de información: De manera individual o grupal, el estudiante deberá buscar información acerca del tema que se esté trabajando. La información como resultado de la búsqueda, puede ser de origen bibliográfico, entrevistas, artículos.

6. Evaluación y calificación

Para la evaluación del ciclo de formación de Programación Básica se debe tener en cuenta:

- Debe ser claro para los beneficiarios que el trabajo en este ciclo es individual.
- La evaluación del ciclo 2 corresponde a la evaluación de los 5 retos y a la evaluación del curso de inglés. Las evaluaciones se deben hacer en la escala 0 a 5.
- La plataforma debe ser capaz de revisar las soluciones entregadas por los beneficiarios y detectar semejanzas que puedan ser interpretadas como copia. En ese caso dichas soluciones se deben revisar individualmente.
- La forma de evaluación del curso de inglés lo define cada operador.
- La nota final del ciclo se calcula así:



- La nota mínima aprobatoria para un ciclo es 3 sobre 5.

Para la evaluación de cada reto se debe tener en cuenta:

- Una evaluación automática de la lógica de negocio, que se hace de una manera similar a la realizada en el ciclo 1. Para esto, al final de cada semana, el beneficiario sube a la plataforma de evaluación su solución para el reto, y la plataforma debe ser capaz de ejecutar las pruebas unitarias y dar una calificación. Esta calificación corresponde al número de requerimientos perfectamente resueltos (que pasen todas las pruebas). La plataforma debe señalar al beneficiario las pruebas que no pasó satisfactoriamente su solución.
- Si la primera parte es exitosa, se hace una evaluación manual, desarrollada por un tutor, de la interfaz de usuario. Para esto se verifica que la interfaz cumpla con todos los requerimientos de funcionamiento definidos en las historias de usuario.
- Si las dos primeras partes son exitosas, se hace una inspección semiautomática del código, en donde se verifican las reglas de buena calidad que debe cumplir un programa orientado por objetos (documentación, encapsulamiento, desacoplamiento, estructura de los algoritmos, entre otros).

7. Cronograma

Semana	Actividades Independientes de Aprendizaje	Actividades Síncronas de Aprendizaje	Tema
1.	Realizar lectura de literatura recomendada Literatura recomendada: [1] – Nivel 1 [2] – Actividad 1, 2, 3 y 4. [3] – Cap. 3 (3.1, 3.2, 3.3, 3.4).	<ul style="list-style-type: none"> • Presentación del curso, profesor y reglas de juego. • Presentación: Problemas y soluciones, comprensión y especificación del problema: El modelo del mundo del problema. • Presentación: Elementos de un programa, paradigma de POO. • Presentación: Tipos de datos, Expresiones y operadores relacionales, lógico, aritméticos y de asignación. • Presentación de la consola como medio de interacción para el usuario. 	<p>Introducción al curso y al paradigma de programación orientada a objetos (POO).</p> <p>Problemas, Soluciones y Programas: Clases, métodos y Objetos.</p> <p>Interacción usuario-computador: la consola.</p>

2.	<p>Realizar lectura de literatura recomendada</p> <p>Literatura recomendada: [1] – Nivel 2, 4 [2] – Actividad 7, 10, 11 19. [3] – Cap. 3 (3.5, 3.8), 11.</p> <p>Solución Reto No. 1</p>	<ul style="list-style-type: none"> • Presentación: Nuevos elementos de modelado. Tipos simples de datos, constantes, asociaciones opcionales. • Presentación: Requerimientos Funcionales • Presentación: Diagramas de clases. Casos de uso, historias de usuario • Presentación: Responsabilidades de una clase. ¿Cómo identificar responsabilidades? • Patrón de diseño MVC: Modelo – Vista – Controlador. • Publicación Reto No. 1 	<p>Introducción al Modelamiento (UML)</p> <p>MVC</p> <p>Patrón GRASP</p>
3	<p>Realizar lectura de literatura recomendada</p> <p>Literatura recomendada: [3] – Capítulo 4 [13] – Nivel 10</p>	<ul style="list-style-type: none"> • Presentación sobre Herencia, clases y métodos abstractos • Presentación sobre Interfaces • Presentación extendida sobre Interfaces y Herencia, complementando con Polimorfismo y diseño de aplicaciones. 	<p>Herencia, clases y métodos abstractos, interfaces, polimorfismo, sobrecarga y sobreescripción de métodos</p>
4	<p>Realizar lectura de literatura recomendada</p> <p>Literatura recomendada: [1] – Nivel 8</p> <p>Solución Reto No. 2</p>	<ul style="list-style-type: none"> • Presentación sobre DAO / DTO • Presentación sobre Excepciones y Diseño de nuevas Excepciones • Presentación documentación de código. • Publicación Reto No. 2 	<p>DAO / DTO</p> <p>Manejo de Excepciones.</p> <p>Documentación de Código</p>
5	<p>Realizar lectura de literatura recomendada.</p> <p>Literatura recomendada: [1] – Nivel 6 [13] – Nivel 2</p> <p>Solución Reto No. 3</p>	<ul style="list-style-type: none"> • Presentación sobre persistencia: Archivos de Texto Plano. • Presentación sobre persistencia: Serialización y Deserialización • Publicación Reto No. 3 	<p>Persistencia con Archivos de Texto, Archivos de Propiedades, Archivos Binarios (Serialización y Deserialización)</p>
6	<p>Realizar lectura de literatura recomendada.</p> <p>Literatura recomendada: [13] – Nivel 6</p> <p>Solución Reto No. 4</p>	<ul style="list-style-type: none"> • Presentación sobre persistencia: Bases Datos • Modelo Entidad Relación • Modelo Relacional • SQL • JDBC • Publicación Reto No. 4 	<p>Persistencia con Bases Datos Conexión entre bases de datos y Java</p>
7	<p>Realizar lectura de literatura recomendada.</p> <p>Literatura Recomendada:</p>	<ul style="list-style-type: none"> • Presentación sobre pruebas de una solución (Pruebas Unitarias y de Usuario), y depuración de programas 	<p>Proceso de Pruebas de una Solución y Depuración de Programas – CRUD & Comparaciones</p>

	<p>[13] – Nivel 1.</p> <p>Solución Reto No. 5</p>	<ul style="list-style-type: none"> • introduciendo el papel de la GUI (Interfaz gráfica de usuario) • Publicación Reto No. 5 	GUI
--	--	---	-----

8. Bibliografía Básica y Complementaria

Básica:

- [1]. Villalobos Salcedo, J. A., Casallas Gutiérrez, R., & Castillo, M. F. (2006). Fundamentos de programación: Aprendizaje activo basado en casos. México: Pearson Educación.
- [2]. Rojas Reales, W., & Silva Montoya, M. (2016). Introducción a java: Guía de actividades prácticas. Bogotá: Universidad El Bosque.
- [3]. Horstmann, Cay S., & Cornell, Gary. (2012). Core Java – Fundamentals. Ed. Prentice Hall.

Complementaria:

- [4]. Allende, J. S. (2013). Programación en java. ES: McGraw-Hill Interamericana - M.U.A.
- [5]. Deitel, H. M., Deitel, P. J., & Romero Elizondo, A. V. (2004). Cómo programar en java (5a ed.). México: Pearson y Prentice Hall
- [6]. Joyanes Aguilar, L., & Zahonero Martínez, I. (2011). Programación en JAVA: Algoritmos, programación orientada a objetos e interfaz gráfica de usuario. México; Bogotá; Buenos Aires: McGraw Hill.
- [7]. Joyanes Aguilar, L., & Zahonero Martínez, I. (2013) Estructuras datos en java. México; Bogotá; Buenos Aires: McGraw Hill.
- [8]. Horstmann, C. S. (2004). Big java: Compatible with java 5, 6 and 7 (4th ed.). Hoboken: John Wiley&Sons.
- [9]. Kak, A. C. (2014;2015;). Designing with objects : Object-oriented design patterns explained with stories from harry potter (1st ed.). New York: John Wiley & Sons, Incorporated.
- [10]. Miller, R., Kasparian, R., & Books24x7, I. (2006). Java for artists: The art, philosophy, and science of object-oriented programming. Falls Church: Pulp Free Press.
- [11]. Preißel, R., & Stachmann, B. (2014). Git: Distributed version control--fundamentals and workflows (1st ed.). Vancouver: Brainy Software.
- [12]. Rossberg, J., & Books24x7, I. (2014). Beginning application lifecycle management (2013;1; ed.). Berkeley, CA: Apress. doi:10.1007/978-1-4302-5813-1
- [13]. Villalobos Salcedo, J. A. (2008). Introducción a las estructuras de datos: aprendizaje activo basado en casos: un enfoque moderno usando Java, UML, objetos y eclipse. Bogotá: Pearson Educación, c 2008.

9. Recursos de aprendizaje

- Aula virtual donde se encuentra todo el material didáctico, guías de ejercicios, soluciones, tareas, etc., correspondiente al desarrollo del ciclo de Programación Básica.
- Computador con acceso a internet para realizar el ingreso a las sesiones síncronas y tutorías, la consulta de la información y la realización de los retos.
- Instalación IntelliJ IDEA y lenguaje JAVA (JDK 1.8)
- Creación de cuenta en Replit (<https://replit.com/>)