



El futuro digital
es de todos

MinTIC

FUNDAMENTOS DE PROGRAMACIÓN



Presentación del curso Programación Básica

GINA PATRICIA FLOREZ CANOSA

Misión
TIC2022



UNIVERSIDAD
EL BOSQUE



Programación Básica:

Se fortalecerán 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



Perfil del Egresado

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. 2

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



El futuro digital
es de todos

MinTIC

FUNDAMENTOS DE PROGRAMACIÓN



Cronograma

Semana	Temas	Evaluación
1	<ul style="list-style-type: none">• Presentación del curso• Problemas y soluciones, comprensión y especificación del problema: El modelo del mundo del problema.• Elementos de un programa, paradigma de POO.• Tipos de datos, Expresiones y operadores relacionales, lógico, aritméticos y de asignación.• Presentación de la consola como medio de interacción para el usuario.	Publicación Reto # 1
2	<ul style="list-style-type: none">• Nuevos elementos de modelado. Tipos simples de datos, constantes, asociaciones opcionales.• Requerimientos Funcionales• Diagramas de clases.• Casos de uso, historias de usuario• Responsabilidades de una clase. ¿Cómo identificar responsabilidades?• Patrón de diseño MVC: Modelo – Vista – Controlador.	Entrega Reto#1 :Domingo 11 de Julio a las 23:55
3	<ul style="list-style-type: none">• Abstracción, Herencia, Polimorfismo, Sobrecarga de métodos, Sobreescritura de métodos.• Clases y Métodos Abstractos• Interfaces	Entrega Reto # 1 (10%)



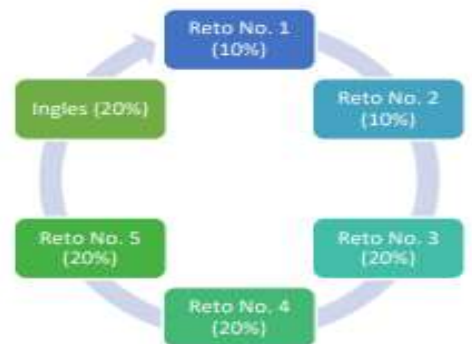
Cronograma

Semana	Temas	Evaluación
4	<ul style="list-style-type: none">• Patrones DAO / DTO• Excepciones y Diseño de nuevas Excepciones• Documentación de código	Publicación del Reto # 2
5	<ul style="list-style-type: none">• Persistencia• Archivos de Texto Plano.• Archivos Binarios.• Serialización y Deserialización	Publicación del Reto # 3 y Entrega del Reto
6	<ul style="list-style-type: none">• Bases Datos: Modelo Entidad Relación y Modelo Relacional• Instrucciones Básicas de SQL• Conexión JDBC	Publicación del Reto # 4 y Entrega del Reto
7	<ul style="list-style-type: none">• Pruebas de una solución (Pruebas Unitarias y de Usuario), y depuración de programas• Introducción a la GUI (Interfaz gráfica de usuario)	Publicación Reto No. 5



Reglas del Juego

1. Sesiones sincrónicas
2. Excusas de inasistencia se envían directamente por el estudiante al grupo de apoyo del proyecto, no al docente. Igual para cualquier otra solicitud.
3. Las entregas de los retos se debe hacer obligatoriamente en el aula de VPL y luego en el aula de Misión TIC. Si el estudiante no carga su reto en la plataforma VPL no se le evaluará.
4. Porcentajes



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



Rúbrica

5. La evaluación se realizará con la reglamentación dada como rúbrica de los retos.

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

- a) 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.
- b) 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.
- c) 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).



El futuro digital
es de todos

MinTIC

FUNDAMENTOS DE PROGRAMACIÓN



Herramientas

1. Se utilizará el IDE Netbeans
 2. La entrega de retos a evaluar será la versión entregada en VPL.
 3. No se realizarán ampliaciones de tiempos de entregas.
 4. Fechas de entrega serán los domingos.
 5. Se realizarán tutorías
-



BIBLIOGRAFÍA

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.