**Programación del Curso - Unidad 1**

**FP:** [Fundamentos de Programación. Aprendizaje Activo Basado en Casos.](https://cupi2.virtual.uniandes.edu.co/images/APO1/fundamentos-de-programacion.pdf) Jorge Villalobos y Rubby Casallas. Universidad de los Andes.

**CPJ**: Cómo Programar en Java. Paul Deitel y Harvey Deitel. Pearson. Décima Edición. Filas de este color son sesiones de Laboratorio. Prácticas en salas de cómputo.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sem** | **Sesión** | **Actividad** | **Lectura Previa** | **Actividad Previa** | **Material Adicional** | **Objetivos Específicos** |
| 1 | U1C1 | **\*Presentación del Curso\***   * [Programa del Curso](https://docs.google.com/document/d/1gDhnHY7f9tr_LqNQv_pwvIFMeAtiP6-0wScOwM2TKWs/edit?usp=sharing) * [Agenda](https://docs.google.com/spreadsheets/d/1HmZYyvRGpBgg1fhHc33d9BwWTYFIwHUhCfgWcT_UJz4/edit?usp=sharing) (muy importante!)   Método de Ingeniería   * [Resumen](https://drive.google.com/file/d/0Bwy19LIX4H2RLThkTV9ZYzhQS0U/view?usp=sharing) * [Presentación](https://drive.google.com/file/d/0Bwy19LIX4H2RQVRnZ2c5NThHbEE/view?usp=sharing)   Introducción a la Programación   * [Motivación](https://drive.google.com/file/d/0Bwy19LIX4H2RLXJYYnV3YUxkWTA/view?usp=sharing) * [Problema (Gráfico)](https://drive.google.com/file/d/0Bwy19LIX4H2RQjQxR2hCcFhIenM/view?usp=sharing) * [Proceso (Gráfico)](https://drive.google.com/file/d/0Bwy19LIX4H2RUjduN3BhV3RveW8/view?usp=sharing) | Deseable: Resumen del Método de la Ingeniería | Deseable: Desarrollo de un mapa conceptual sobre el Método de la Ingeniería | * [Presentación Motivación](https://drive.google.com/open?id=1NCS5048ADUmpnQRuavuGqcPvxH3nPC1rLOib6smTtfg) |  |
| 1 | U1C2 | Problemas y Soluciones   * Especificación de un Problema. [Análisis del problema](https://drive.google.com/file/d/0Bwy19LIX4H2RenRpcGZXS3R2LVU/view?usp=sharing) * El Proceso y las Herramientas. [Solución de un problema con un programa de computador](https://drive.google.com/file/d/0Bwy19LIX4H2RWGp3eU1WaG9lZnM/view?usp=sharing)   **\*Requerimientos Funcionales\*** | [FP: Introducción y Requerimientos Funcionales. Desde la página 19 hasta la página 41](https://drive.google.com/file/d/0Bwy19LIX4H2RUTA1UlJVUXczclE/view?usp=sharing) | Responda:   * Qué es la especificación de un problema? * Qué es un requerimiento funcional? * Cuáles son las etapas del proceso de desarrollo de software? * Indique y explique cada uno de los campos que hacen parte del formato para especificar un requerimiento funcional. | * [Presentación RF y RNF](https://drive.google.com/open?id=1W60rzrE261wntVJq9FBGJt1Y0Hr_vDN2vUadeFOLLjU) | OE1.1. Explicar el proceso global de solución de un problema usando un programa de computador. Esto incluye las etapas que debe seguir para resolverlo y los distintos elementos que debe ir produciendo a medida que construye la solución. |
| 1 | U1Lab1 | Introducción a Java   * [Construcción de un programa (Gráfico)](https://drive.google.com/file/d/0Bwy19LIX4H2RZDBNZ0hSTW91UHM/view?usp=sharing) * [Entorno de Desarrollo Típico en Java (Gráfico)](https://drive.google.com/file/d/0Bwy19LIX4H2RUUJRSTU5SXJxMlk/view?usp=sharing) * [Enunciado del Laboratorio 1](https://docs.google.com/document/d/1Rl6AB-JnfKdUOulJSoc8UpibpG4Rci3cVdOs5AJk-IE/edit?usp=sharing) * Rúbrica laboratorio 1 | [FP: Qué es un Algoritmo. Desde la página 61 hasta la página 63 (hasta antes de 6.2. Clases y Objetos)](https://drive.google.com/file/d/0Bwy19LIX4H2RNzY2Zms3ZW90SFU/view?usp=sharing)  [CPJ: Introducción a Java. Desde la página 34 hasta la página 54](https://drive.google.com/file/d/0Bwy19LIX4H2RMnFlQ3NLYzhSaVk/view?usp=sharing) | * Descargar el JDK (Java Development Kit) del sitio oficial: <http://www.oracle.com/technetwork/java/javase/downloads> * [Lectura Opcional. CPJ: Introducción a las computadoras. Desde la página 1 hasta la página 33](https://drive.google.com/file/d/0Bwy19LIX4H2RMUprODRBSUctc00/view?usp=sharing) | * [Mi primer Programa](https://docs.google.com/document/d/1VejRApG_wB3ObJV9oLrcvEsi6J5U6lC9L-UqojVboa8/edit?usp=sharing) | OE1.4. Explicar el concepto de algoritmo y su importancia para la solución de problemas.  OE1.6. Utilizar un ambiente de desarrollo de programas (incluyendo la compilación y ejecución de programas desde consola) y un espacio de trabajo predefinido, para construir la solución de un problema. |
| 2 | U1C3 | El Modelo del Mundo del Problema | [FP: El Modelo del Mundo del Problema. Desde la página 42 hasta la página 60](https://drive.google.com/file/d/0Bwy19LIX4H2RSHBQdVltSktpaGs/view?usp=sharing) | Responda:   * Qué es una entidad? Qué es una característica? * Qué es una relación? Qué es UML? * Cuál es la diferencia entre requerimientos funcionales y no funcionales? | * [Presentación Modelo del Mundo](https://drive.google.com/open?id=1fFK3SGWGKyhJWrJxx0GCrEJs_Hi3p-lARKOd3BBcGO0) | OE1.2. Analizar un problema que se va a resolver usando un programa de computador, construyendo un modelo con los elementos que intervienen en el problema y especificando los servicios que el programa debe ofrecer. |
| 2 | U1C4 | Clases y Objetos   * [Clase (Gráfico)](https://drive.google.com/file/d/0Bwy19LIX4H2RZ04yUDdEZGg4aFE/view?usp=sharing) | [FP: Clases y Objetos. Desde la página 63 hasta la página 68](https://drive.google.com/file/d/0Bwy19LIX4H2ReGFMUXdZTFVuMWc/view?usp=sharing)  [CPJ: Introducción al POO. Desde la página 10 hasta la página 13](https://drive.google.com/file/d/0Bwy19LIX4H2RbTJiV2tlYWQ3RUk/view?usp=sharing) | Responda:   * Qué es una clase? * Qué es un objeto? * Qué diferencia hay entre clases y objetos? * Todo diagrama de objetos está asociado a un diagrama de clases? Por qué? | * [Presentación cómo codificar el modelo del mundo](https://drive.google.com/open?id=1Ggu92pyuzlBoIAOm1Z3b5l54jDQobF_h35ted07FblA) * [Presentación clase vs Objeto](https://docs.google.com/presentation/d/1bxbS0Gl8YToflV8dKcnCGnZt6v5zyZStpCkbC_vB-wk/edit?usp=sharing) * [Video Motivación: Clases y Objetos](https://www.youtube.com/watch?v=8UVXm22U7vQ) * [Video Introducción: Clases y Objetos](https://www.youtube.com/watch?v=YxJPswUKGek) |  |
| 2 | U1Lab1 | * Java como Lenguaje de Programación Orientado a Objetos (POO) * Objetos para encapsular atributos. | [FP: Java como LPOO. Desde la página 68 hasta la página 77](https://drive.google.com/file/d/0Bwy19LIX4H2RTTdSOHJYTDE3cGc/view?usp=sharing) |  | [Ejemplo diagrama de Objetos](https://drive.google.com/open?id=0B5d7IWvwA3L2YzhqVWsxdXptWTQ) | OE1.8. Aplicar el concepto de encapsulamiento en el modelamiento e implementación de soluciones bajo el paradigma de programación orientado a objetos.  OE1.11. Construir las clases que implementan el modelo de la solución del problema. |
| 3 | U1C5 | Métodos | [FP: Métodos. Desde la página 77 hasta la página 95](https://drive.google.com/file/d/0Bwy19LIX4H2RTDRsUTNjemZqaW8/view?usp=sharing)  [CPJ: Introducción a POO en Java Parte 1. Desde la página 69 hasta la página 81](https://drive.google.com/file/d/0Bwy19LIX4H2RN05KRVVlNlpXeDg/view?usp=sharing) | * ¿Qué es un tipo de dato? Describa la información que se puede representar con los tipos de datos: int, double, String * Identificar las partes que componen un método (signatura, cuerpo) |  | OE1.5. Utilizar expresiones simples: declaraciones de variables, asignaciones e invocaciones a métodos. Esto implica entender los conceptos de parámetro y de creación de objetos.  OE1.9. Identificar de manera informal los métodos de una clase y clasificarlos en métodos para inicializar, consultar y modificar.  OE1.10. Declarar y utilizar variables diferenciando entre tipos de datos primitivos y tipos de datos referenciados. |
| 3 | U1C6 | Diseño de la Solución | **FPx3**  [FP: Diseño Solución. Desde la página 96 hasta la página 99](https://drive.google.com/file/d/0Bwy19LIX4H2RRGdKYVlZOTNDcFE/view?usp=sharing)  [FP: Clases, Objetos y Métodos. Desde la página 176 hasta la página 184](https://drive.google.com/file/d/0Bwy19LIX4H2RbTdGOElZU2NIY28/view?usp=sharing)  [FP: Tipos de Métodos. Desde la página 197 hasta la página 207](https://drive.google.com/file/d/0Bwy19LIX4H2RVmxfdDZZX2U0SlU/view?usp=sharing)  [CPJ: desde la página 81 hasta la página 100](https://drive.google.com/file/d/0Bwy19LIX4H2ReHdvSk5TYlMzS3c/view?usp=sharing) | * ¿Cuáles son los tipos de método según la operación que realizan? Explique brevemente cada tipo. * ¿Cuáles son las estrategias para identificar responsabilidades? Explíquelas. | [Mapa mental métodos](https://drive.google.com/file/d/0B3iu5fjawCjuNXBPX0NBU1BSSVk/view?usp=sharing)  [Instrucción new](https://drive.google.com/file/d/0B3iu5fjawCjucU5zQ0xCUmd2NWs/view?usp=sharing) | OE1.3. Explicar la estructura de un programa de computador y el papel que desempeña cada uno de los elementos que lo componen.  OE1.13. Utilizar una arquitectura para un programa que permita repartir de manera adecuada las responsabilidades entre la interfaz de usuario y el modelo de la solución. El estudiante deberá poder explicar la importancia de mantener separadas las clases de estos dos dominios. |
| 3 | U1Lab1 | * **Entrega del Laboratorio 1** | Seguiremos trabajando en la construcción con un editor de texto simple y compilando por Consola. |  |  | OE1.7. Interpretar la documentación (modelos conceptuales y especificación de las operaciones) de una solución con el fin de: a) llevar a cabo la implementación con base en el diseño (rol del programador vs rol del diseñador), b) comprender el desarrollo en su totalidad, c) verificar la coherencia entre análisis, diseño e implementación (trazabilidad de los elementos en los artefactos del desarrollo) y, d) validar la solución propuesta en el diseño con las necesidades del usuario expuestas en los requerimientos funcionales.  OE1.12. Construir las clases que implementan una interfaz de usuario e integrarlas con las clases que implementan el modelo de la solución del problema. |