



## Ciclo 2

# Semana 1

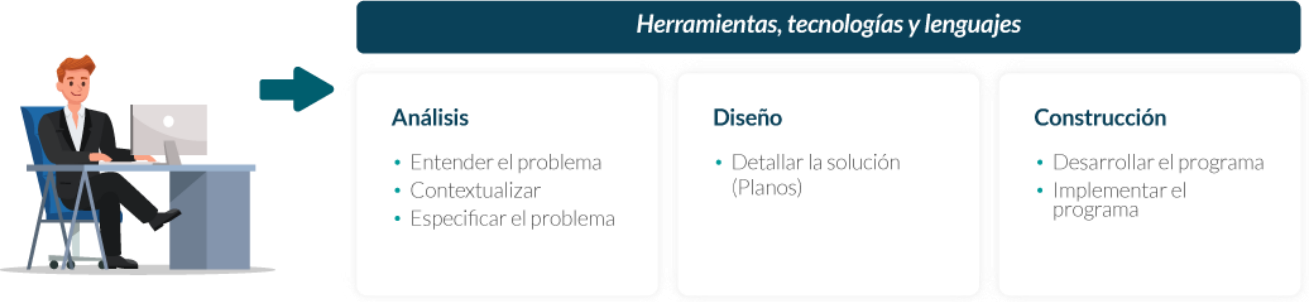
## Introducción a la Programación Orientada a Objetos (POO)

### Lectura 1 - Etapas en la solución de problemas

## Etapas en la solución de problemas



Entender y detallar un problema es sólo la primera parte dentro del desarrollo de un programa. A continuación, se presentan tres grandes etapas que el programador realizará:



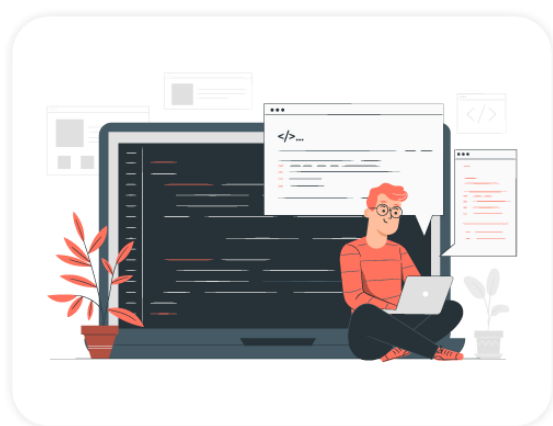


## Análisis



Lo primero que debe hacer nuestro programador es iniciar con un proceso de construcción de software, definiendo el problema de una manera tan clara y detallada que cualquier persona de su equipo de trabajo pueda entenderlo, todo esto a partir de la información dada por el cliente. Esta primera etapa se conoce como análisis y el resultado o salida de esta se conocerá como especificación del problema. Así al analizar un problema se tratará de garantizar que la solución dada si resuelva lo solicitado y no algo diferente al problema que tiene el cliente.

**En general todo problema puede ser dividido entre los siguientes 3 aspectos:**

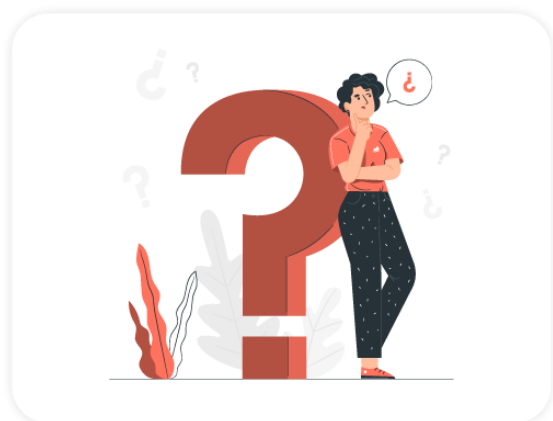


**Requerimientos funcionales:** acciones o servicios que el programa debe proveer al usuario. Este aspecto hace referencia a lograr identificar cuál es ese servicio que el cliente espera que el programa prestará.

Este se representa con cuatro elementos como lo son un identificador y un nombre, el resumen de la operación, entradas que el usuario provee para que el programa realice la operación indicada y finalmente el resultado que pueden ser una modificación de un valor, el cálculo de un valor, o la combinación de estos dos.



## Semana 1

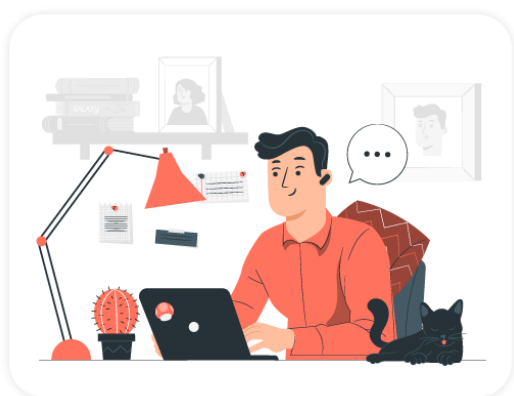
Introducción a la Programación  
Orientada a Objetos (POO)

**Mundo del problema:** ¿contexto en el que ocurre un problema! Al programador no debe bastarle con saber y entender la funcionalidad que tendrá su programa sino debe entender en qué contexto se ejecutará el mismo, como por ejemplo la estructura y el funcionamiento de la empresa.

Para que el programador construya este mundo del problema debe primero identificar las entidades del mundo que vienen siendo elementos que intervienen en el problema, ya

sean concretos como por ejemplo una persona, un edificio, o abstractos o no tangible como lo es una cuenta bancaria.

En la programación orientada a objetos, estas entidades son conocidas como clases y son pieza fundamental en el diseño, construcción e implementación de cualquier programa. Una vez se definen las entidades de mundo, es necesario documentar las características de estas para poder definir sus relaciones y finalmente documentar información adicional como restricciones o reglas que se presenten sobre dichas entidades.



**Requerimientos no funcionales:** restricciones o condiciones que indica el cliente para el programa que se le va a construir, y corresponden a todas estas reglas restrictivas o condiciones que el cliente manifiesta y que se deben cumplir o tener en cuenta, y en gran parte dependen del contexto del programa como lo son restricciones en tecnología a usar, cantidad de usuarios, entre otros.





**El programador no puede resolver un problema que no entiende.**

→ 1ra tarea: entender el problema



#### Elementos de especificación:

- Los requerimientos del usuario.
- El mundo del problema.
- Las restricciones del cliente.

#### Pensemos en la construcción de una carretera

- Carriles que debe tener la carretera
- El tipo de terreno, humedad, ubicación
- Presupuesto máximo disponible.

Al tener en cuenta estos tres elementos de especificación se facilitará la tarea de entender el problema y se simplificará así su análisis. Al final de la etapa de análisis debe plasmarse todo en un documento que pueda ser revisado por el equipo de trabajo y por el cliente para validar correctamente nuestro trabajo.

## Diseño



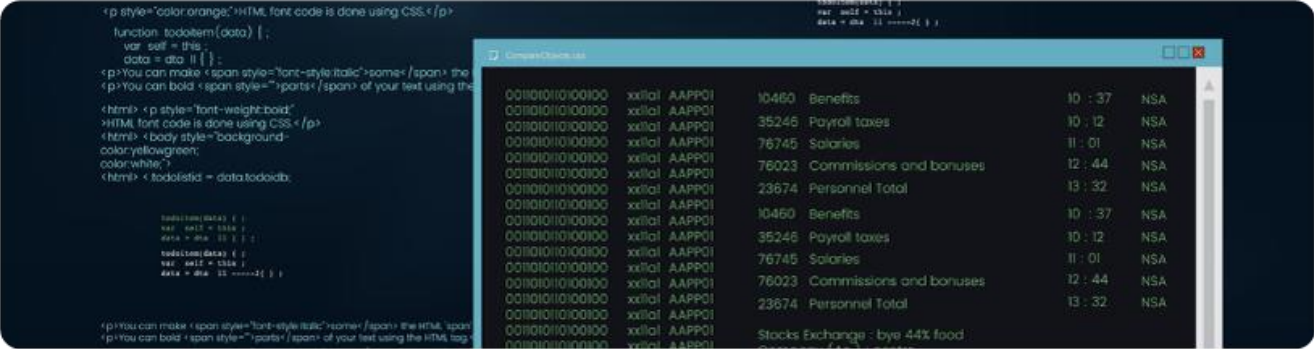
Semana I

Introducción a la Programación Orientada a Objetos (POO)

En esta etapa el programador definirá su estrategia, imaginando una posible solución y definiendo sus componentes.

El objetivo principal es detallar todas las características que tendrá nuestra solución para una vez detallada empezar con la construcción del software. Aquí se usarán herramientas de apoyo como dibujos, ecuaciones, planos, diagramas, documentos entre otros.

Construcción de la solución



En esta última etapa se tiene como objetivo la implementación del programa que fue desarrollado a partir de un diseño previo y el cual se probará para evidenciar que funciona de manera correcta.

El ciclo de vida de construcción



Dar una solución a un problema implica componentes como el diseño y sus planos de solución que definen la estructura del programa. Estos planos serán fundamentales en el momento de hacer mantenimientos o cambios al mismo. Un segundo componente es el código fuente escrito en el lenguaje de programación utilizado y este se crea al utilizar el ambiente de desarrollo seleccionado y en el mismo se puede editar.

Por último, se cuenta con los archivos de construcción del programa en donde se explica cómo usar el código fuente para crear un código que sea ejecutable y que será instalado y ejecutado por un compilador en los dispositivos de comunicación final con el usuario.

Villalobos en su libro Fundamentos de programación: Aprendizaje activo basado en casos, nos muestra con dos ilustraciones muy explicativas el mapa conceptual de la solución de un problema y el de la construcción de un programa:<sup>1</sup>

<sup>1</sup> Villalobos Salcedo, J. A., Casallas Gutiérrez, R., & Castillo, M. F. (2006). Fundamentos de programación: Aprendizaje activo basado en casos. México: Pearson Educación

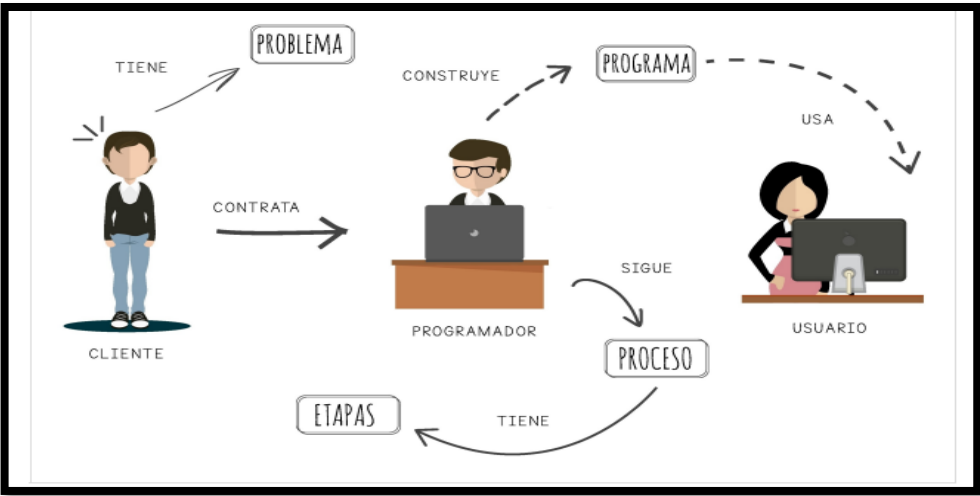


Figura 1: mapa conceptual de la solución de un problema con un computador

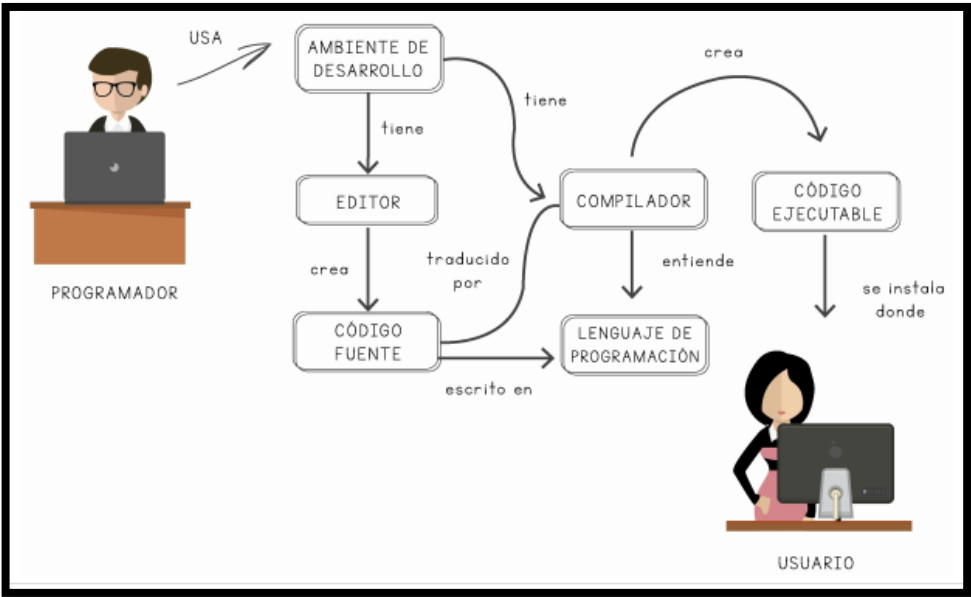


Figura 2: mapa conceptual de la construcción de un programa