

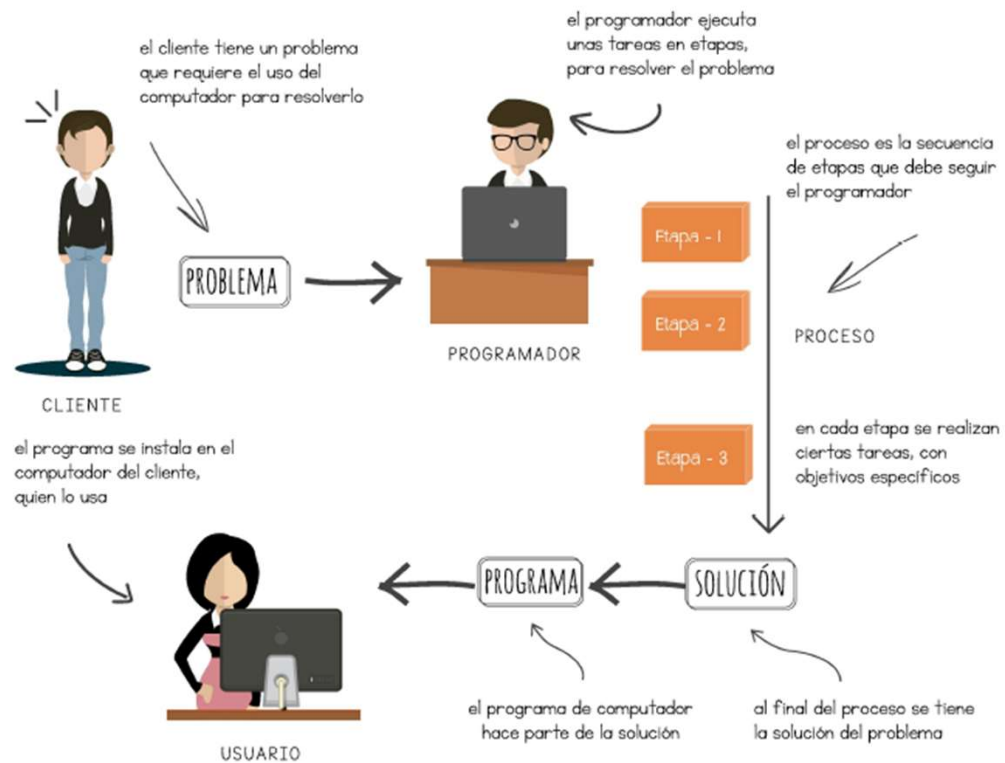
# Fundamentos de Programación

---

JUAN G. CORTES V.



# Proceso de solución de un problema

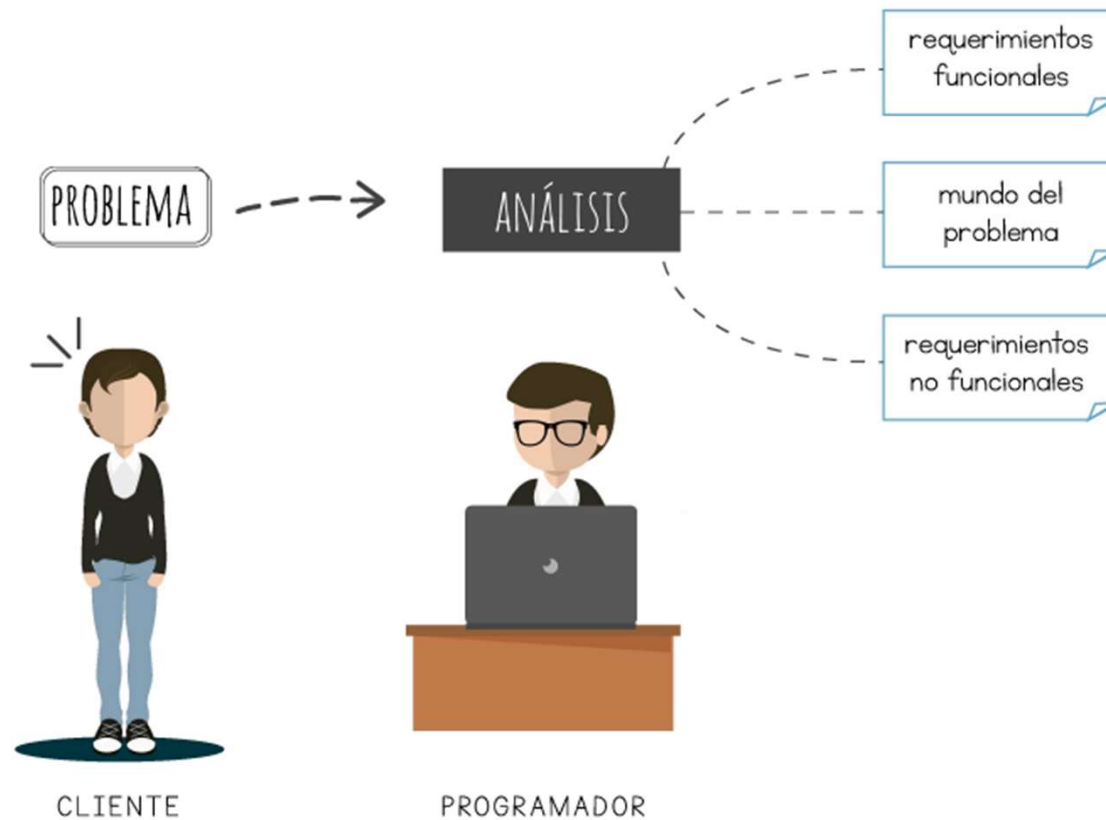


---

# ANALISIS DEL PROBLEMA



# Análisis del problema



# Análisis del problema

---

“Analizar un problema es tratar de entenderlo. Esta etapa busca garantizar que no tratemos de resolver un problema diferente al que tiene el cliente.

Descomponer el problema en sus tres aspectos fundamentales, facilita la tarea de entenderlo: en cada etapa nos podemos concentrar en sólo uno de ellos, lo cual simplifica el trabajo.

Esta descomposición se puede generalizar para estudiar todo tipo de problemas, no sólo se utiliza en problemas cuya solución sea un [programa de computador](#).

Además de entender el problema, debemos expresar lo que entendemos siguiendo algunas convenciones.

Al terminar la etapa de [análisis](#) debemos generar un conjunto de documentos que contendrán nuestra comprensión del problema. Con dichos documentos podemos validar nuestro trabajo, presentándoselo al cliente y discutiendo con él.”

# Ejemplo 1: Identificar los aspectos que hacen parte de un problema


El problema: una empresa de aviación quiere construir un programa que le permita buscar una ruta para ir de una ciudad a otra, usando únicamente los vuelos de los que dispone la empresa. Se quiere utilizar este programa desde todas las agencias de viaje del país.

Cliente	La empresa de aviación.
Usuario	Las agencias de viaje del país.
Requerimiento funcional	R1: dadas dos ciudades C1 y C2, el programa debe dar el itinerario para ir de C1 a C2, usando los vuelos de la empresa. En este ejemplo sólo hay un <b>requerimiento funcional</b> explícito. Sin embargo, lo usual es que en un problema haya varios de ellos.
Mundo del problema	En el enunciado no está explícito, pero para poder resolver el problema, es necesario conocer todos los vuelos de la empresa y la lista de ciudades a las cuales va. De cada vuelo es necesario tener la ciudad de la que parte, la ciudad a la que llega, la hora de salida y la duración del vuelo. Aquí debe ir todo el conocimiento que tenga la empresa que pueda ser necesario para resolver los requerimientos funcionales.
Requerimiento no funcional	El único <b>requerimiento no funcional</b> mencionado en el enunciado es el de distribución, ya que las agencias de viaje están geográficamente dispersas y se debe tener en cuenta esta característica al momento de construir el programa.

Requisitos Funcionales	Requisitos No Funcionales
Describen qué hace el sistema.	Describen cómo debe ser el sistema.
Detallan las funcionalidades específicas que el sistema debe proporcionar.	Definen las cualidades, características y restricciones del sistema.
Se centran en las funciones y el comportamiento del sistema.	Se centran en la calidad, el rendimiento, la seguridad y las características del sistema.
Orientados a lo que el sistema debe lograr en términos de comportamiento y resultados.	Orientados a cómo el sistema debe cumplir con los requisitos de calidad, rendimiento y experiencia.
Medidos por su capacidad para cumplir con escenarios y casos de uso específicos.	Medidos por atributos como velocidad, seguridad, usabilidad, disponibilidad, entre otros.
Directamente afectan la interacción del usuario con el sistema.	Indirectamente afectan la experiencia del usuario y la calidad del sistema.
Se traducen en funcionalidades específicas del sistema.	Se traducen en criterios de rendimiento, seguridad, usabilidad, entre otros.

---

# Requerimientos funcionales vs no funcionales



# Requerimientos funcionales vs no funcionales

---

## Requerimiento funcional

Estos requisitos especifican lo que debe hacer el sistema. Describen las interacciones entre el software y sus usuarios, así como el comportamiento del software bajo diversas condiciones.

## Requerimiento No funcional

Se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema (características de usuario), sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad. En palabras más sencillas, no hablan de “lo que” hace el sistema, sino de “cómo” lo hace.



# Requerimientos funcionales

---

Un requerimiento funcional se describe a través de cuatro elementos

Un identificador y un nombre.

Un resumen de la operación.

Las entradas (datos) que debe dar el usuario para que el programa pueda realizar la operación.

El resultado esperado de la operación. Hay tres tipos posibles de resultado en un **requerimiento funcional**: (1) una modificación de un valor en el mundo del problema, (2) el cálculo de un valor, o (3) una mezcla de los dos anteriores.

# Plantee las siguientes tres preguntas

---

1. Poder realizar esta operación es una de las razones por las cuales el cliente necesita construir un programa?

Esto descarta todas las opciones que están relacionadas con el manejo de la interfaz ("poder cambiar el tamaño de la ventana", por ejemplo) y todos los requerimientos no funcionales, que no corresponden a operaciones sino a restricciones.

2. ¿La operación no es ambigua?

3. ¿La operación tiene un comienzo y un fin?

Un **requerimiento funcional** se puede ver como un servicio que el programa le ofrece al usuario para resolver una parte del problema.

# Requerimientos no funcionales

---

Se establecen para especificar criterios que evalúen la operación de un servicio y se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento:

**Eficiencia:** se relaciona con el desempeño del sistema e implica la cantidad de operaciones o respuestas ante un tiempo determinado

**Usabilidad:** esfuerzo necesario para que un usuario aprenda a usar e interpretar el sistema, entorno o producto.

**Escalabilidad:** propiedad de un sistema para adaptarse ante el crecimiento y la demanda cambiante

**Portabilidad:** propiedad de un sistema, objeto o programa para llevarse, usarse o ejecutarse en cualquier parte.

# Caso de estudio

Para este caso de estudio vamos a considerar un programa que administra la información de un empleado.

El empleado tiene un nombre, un apellido, un género (masculino o femenino), una fecha de nacimiento y una imagen asociada (su foto). Además, tiene una fecha de ingreso a la empresa en la que trabaja y un salario básico asignado.

Desde el programa se debe poder cambiar el salario del empleado, lo mismo que hacer los siguientes cálculos con la información disponible: (1) edad actual, (2) antigüedad en la empresa y (3) prestaciones a las que tiene derecho en la empresa.

The screenshot shows a software window titled "Sistema de Empleados". It contains three main sections: "Datos Personales", "Salario", and "Cálculos".

- Datos Personales:** Includes input fields for "Nombre:" (Pedro), "Apellido:" (Matalana), "Sexo:" (m), "Fecha de Nacimiento:" (16-6-1982), and "Fecha de Ingreso:" (5-4-1997). To the right is a placeholder image of a person with a sad face.
- Salario:** Shows a "Salario:" field with the value "\$2.300.000" and a "Modificar" button.
- Cálculos:** Contains three buttons and corresponding output fields:
  - "Calcular Edad" with the result "33".
  - "Calcular Antigüedad" with the result "18".
  - "Calcular Prestaciones" with the result "\$3.450.000".

Nombre	<b>R1:</b> actualizar el salario básico del empleado	<b>Es conveniente asociar un identificador con cada requerimiento, para poder hacer fácilmente referencia a él. En este caso el identificador es R1. Es aconsejable que el nombre de los requerimientos corresponda a un verbo en infinitivo, para dar una idea clara de la acción asociada con la operación. En este ejemplo el verbo asociado con el requerimiento es "actualizar".</b>
Resumen	Permite modificar el salario básico del empleado	El resumen es una frase corta que explica sin mayores detalles el <a href="#">requerimiento funcional</a> .
Entradas	Nuevo salario	Las entradas corresponden a los valores que debe suministrar el usuario al programa para poder resolver el requerimiento. En el requerimiento del ejemplo, si el usuario no da como entrada el nuevo salario que quiere asignar al empleado, el programa, no podrá hacer el cambio. Un requerimiento puede tener cero o muchas entradas. Cada entrada debe tener un nombre que indique claramente su contenido. No es buena idea utilizar frases largas para definir una entrada.
Resultado	El salario del empleado ha sido actualizado con el nuevo salario	El resultado del <a href="#">requerimiento funcional</a> de este ejemplo es una modificación de un valor en el mundo del problema: el salario del empleado cambió. Un ejemplo de un requerimiento que calcula un valor podría ser aquél que informa la edad del empleado. Fíjese que el hecho de calcular esta información no implica la modificación de ningún valor del mundo del problema. Un ejemplo de un requerimiento que modifica y calcula a la vez, podría ser aquél que modifica el salario del empleado y calcula la nueva retención en la fuente.

# Elementos de un requerimiento funcional

# SMARTWATCH

## FUNCIONALES

- El usuario podrá parear el dispositivo a través de un código numérico o un código QR.
- El programa calculara la saturación de oxígeno y los pulsos cardiacos.
- El sistema recibirá notificaciones de correo electrónico y WhatsApp.



## NO FUNCIONALES

- El dispositivo será portable.
- El dispositivo será ergonómico.
- Los materiales presentan durabilidad (vida útil larga bajo condiciones óptimas de funcionamiento).
- El programa será eficiente

# Referencias

---

Villalobos J. Fundamentos de Programación. Universidad de los Andes

Ruiz V. Manual de análisis y diseño de algoritmos. INACAP

Cisco. Fundamentos de Python 1.

Cartagena99. Fundamentos de programación

<https://view.genially.com/65168bf30af28d0011da791d/horizontal-infographic-diagrams-proceso-de-compilacion>

[https://colab.research.google.com/drive/1xcBPcHlVT5gPOe2xPNYjcAPGDS1HKxa3#scrollTo=giq15p\\_ANzE](https://colab.research.google.com/drive/1xcBPcHlVT5gPOe2xPNYjcAPGDS1HKxa3#scrollTo=giq15p_ANzE)

[https://intef.es/observatorio\\_tecno/pseint-programando-en-pseudocodigo/](https://intef.es/observatorio_tecno/pseint-programando-en-pseudocodigo/)

<https://basesdeda2s.data.blog/2019/03/26/tipos-de-datos-y-expresiones/>

<https://platzi.com/clases/3221-pensamiento-logico/50678-operadores-aritmeticos/>

[https://colab.research.google.com/drive/1ytmjdw3qM29er\\_4pUxbadJtC5nSXmVrX?usp=sharing](https://colab.research.google.com/drive/1ytmjdw3qM29er_4pUxbadJtC5nSXmVrX?usp=sharing)