



Análisis de casos

¿Qué es el análisis de casos?



El análisis consiste en descomponer un todo en sus partes, con el fin de comprender la forma en que el todo se constituye y se comporta.

El análisis tiene como finalidad lo siguiente:

1. Determinar de forma clara los requerimientos del cliente
2. Determinar el alcance del programa

3. Determinar la interfaz y el comportamiento esperado del programa

Determinar de forma clara los requerimientos del cliente.



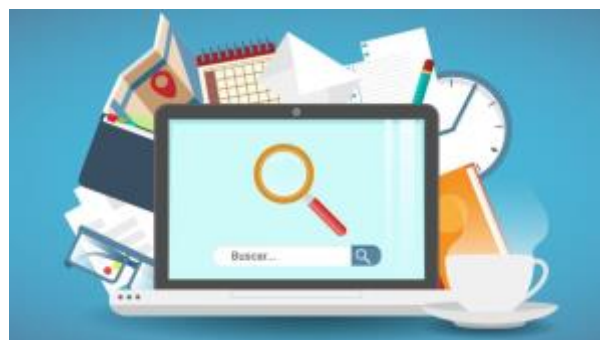
Consiste en llegar a **conocer y documentar** lo que el cliente espera de nuestra aplicación. Es indispensable conocer qué espera el cliente, dado que nuestro éxito se alcanzará cuando se cumpla con la necesidad que éste valora.

Preparación previa.

El proceso para adquirir conocimiento de una disciplina es:

1. Identifique la disciplina pertinente al problema
2. Conozca la terminología especializada
3. Conozca las teorías y principios que fundamentan la disciplina
4. Conozca los procesos inherentes a la disciplina.

Una vez que se tengan los términos relacionados, podremos identificar que teorías de la disciplina nos pueden ser útiles y aplicables.



Procedimiento propuesto para el análisis.

1. Investigar el área del conocimiento a la que pertenece el problema.
2. Documéntese en el área del conocimiento del problema, hágalo antes de acudir con el cliente a resolver dudas.
3. Vea el todo y trate de resolverlo.
4. Separe las cosas que si entiende, de las que no entiende.
5. Enumere las cosas que no entiende, y pregúntelas a quien si sepa
6. Comience preguntando las cosas que le sugieren alguna idea, aunque sea vaga o imprecisa y deje para el final aquellas de las que no entienda en lo absoluto
7. Pregunte siempre cuál es la relación de las cosas entre ellas.



Validación de requerimientos.

Elabore un informe que detalle los requerimientos recopilados y preséntelo al cliente para su validación.



Determinar el alcance del programa



El alcance de un programa consiste en la cantidad de funcionalidad mínima para que el cliente considere a la aplicación como un producto terminado.

Las unidades de funcionalidad son aquellos resultados y comportamientos específicos que el cliente desea que el programa cumpla. La fase de pruebas consistirá, en realidad, en verificar que todas las unidades de funcionalidad estén presentes (Algo que el cliente identifica como una actividad que será posible desarrollar con la aplicación que, si bien puede ser independiente y autónoma, se integra como parte de un todo.).

Se recomienda que cada unidad de funcionalidad tenga:

1. Un identificador
2. Una prioridad
3. Una obligatoriedad

Es muy importante en un proyecto establecer que se cumplirá con aquellas funciones acordadas, ya que esto le permitirá calcular el tiempo y el costo del desarrollo.

Determinar la interfaz y el comportamiento de un sistema.

Es necesario que el cliente tenga al menos una idea de la forma en que el programa lucirá, existe toda una disciplina especializada en este tema a la cual se le conoce como *"usabilidad"*.

Interfaz

Es el medio a través del cual el usuario y la aplicación se comunican.

Una interfaz debe ser:

1. Sencilla de entender
2. Fomentar su uso intuitivo

3. Apegarse a los estándares del ambiente
4. Validar los dominios
5. Tener una secuencia lógica de sus elementos
6. Preguntar la cantidad mínima, suficiente y necesaria para el procesamiento o almacenamiento.



Simulación

Es una representación breve del comportamiento que esperamos del programa, incluso puede hacerse de forma documental. Tiene el propósito de mostrar la forma en que el programa responde ante unos determinados datos de entrada. Para esto es muy importante que el usuario proporcione un ejemplo de los resultados correctos que el programa deberá producir considerando unos datos de entrada determinados.

Si el tiempo lo permite, la simulación se puede realizar a través de una aplicación “mock”. También existen recursos que permiten visualizar de manera simplificada incluso aplicaciones gráficas (<https://www.mockflow.com/>).

A esta versión no definitiva se le llama *prototipo*.

