**Diseño**

En el diseño modelamos el sistema y encontramos su forma para que soporte todos los requisitos, incluyendo los requisitos no funcionales y otras restricciones.

El modelo de análisis proporciona una comprensión detallada de los requisitos y lo que es más importante, impone una estructura del sistema que debemos esforzarnos por conservar lo más fielmente posible cuando demos forma al sistema

Los propósitos del diseño son:

* Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes r
* Crear una ente utilizables, sistemas operativos, tecnologías de distribución y concurrenciarada apropiada y un punto de partida para actividades de implementación subsiguiente capturando los requisitos o subsistemas individuales, interfaces y clases
* Ser capaces de descompones los trabajo de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo

**Artefacto**

* **Modelo de diseño**

El modelo de diseño es un modelo de objetos que describe la realización física de los CU centrándose en como los requisitos funcionales y no funcionales junto con otras restricciones relacionadas con el entorno de implementación, tiene un impacto en el sistema a considerar

En el modelo de diseño, los CU son realizados por las clases de diseño y sus objetos. Esto se representa por colaboraciones en el modelo de diseño y denota realización de casos de uso- diseño. Adviértase que realización de caso de uso- diseño es diferente de la realización de casos de uso-análisis. Lo anterior describe como se realiza un caso de uso en términos de interacción entre objetos del diseño, mientras que lo último describe como se realiza un caso de uso en términos de interacción entre objetos del análisis

* **Clases del diseño**

Una clase de diseño es una abstracción sin costura de una clase o construcción similar en la implementación del sistema. Esta abstracción es sin costura en el siguiente sentido:

* + El lenguaje utilizado para especificar una clase del diseño es lo mismo que el lenguaje de programación.
  + La visibilidad de los atributos y las operaciones de una clase de diseño se especifica con frecuencia. Por ejemplo, las palabras clave, public, protected, private son muy usadas en C++.
  + Las relaciones de aquellas clases del diseño implicadas con otras clases, a menudo tienen un significado directo cuando la clase es implementada.
  + Los métodos de una clase del diseño tiene correspondencia directa con el correspondiente método en la implementación de las clases
  + Una clase de diseño menudo aparece como un estereotipo sin costuras que se corresponde con una construcción en el lenguaje de programación dado. Por ejemplo, una clase de diseño para una aplicación en Visual Basic podría estereotiparse como un <<class module>>, <<form>>
* **Realización de caso de uso-diseño**

Una realización de caso de uso- diseño es una colaboración en el modelo de diseño que describe como se realiza un caso de uso específico y como se ejecuta en términos de clases de diseño y sus objetos

Una realización de casos de uso- diseño tiene una descripción de flujos de eventos textuales, diagramas de clases que muestren sus clases de diseño participantes y diagramas de interacción que muestren la realización de un flujo del diseño

Una realización de caso de uso- diseño proporciona una realización física de la realización de caso de uso-análisis para la que es trazado y también gestiona muchos requisitos no funcionales capturados de la realización de casos de uso-análisis

**Diagramas**

* + **Diagramas de clases**
  + **Diagrama de interacción**
  + **Flujo de sucesos-diseño**

Los diagramas de realización de caos de uso y especialmente los diagramas de interacción son difíciles de interpretar por si solos, Por esto, puede ser útil el artefacto de flujo-de-sucesos- diseño, que es una descripción textual que explica y complementa a los diagramas y sus etiquetas

* + **Requisitos de implementación**

Los requisitos de la implementación son una descripción textual que recoge requisitos, tales como los requisitos no funcionales, sobre una realización de CU. Se refiere a requisitos que se capturan solo en la fase de diseño, pero que es mejor tratar en la implementación.

* **Subsistema de diseño**

Los subsistemas de diseño son una forma de organizar los artefactos del modelo de diseño en piezas más manejables. Un subsistema puede constar de clases del diseño, realizaciones de casos de uso, interfaces y otros

* **Interfaz**

Las interfaces se utilizan para especificar las operaciones que proporcionan las clases y los subsistemas del diseño

Un subsistema que proporcione una interfaz debe contener también clases del diseño u otros subsistemas que proporcionen la interfaz

Las interfaces constituyen una forma de separar la especificación de la funcionalidad en términos de operaciones de sus implementaciones en términos de métodos

* **Descripción de la arquitectura(Vista del modelo de diseño)**

La descripción de la arquitectura contiene una vista de la arquitectura del modelo de diseño que muestra sus artefactos relevantes para la arquitectura

* **Modelo de despliegue**

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre nodos de computo

* **Descripción de la arquitectura (Vista del modelo de despliegue)**

La descripción de la arquitectura contiene una vista de la arquitectura del modelo de despliegue que muestra sus artefactos relevantes para la arquitectura.

**Flujo de trabajo**

**Actividades**

* **Diseño de la arquitectura**

El objetivo del diseño la arquitectura es esbozar los modelos de diseño y despliegue y su arquitectura mediante la identificación de los siguientes elementos:

* + Nodos y sus configuraciones de red(Diagrama de despliegue)
  + Subsistemas de sus interfaces(identificación de subsistemas)
  + Clases del diseño significativas para la arquitectura, como las clases activas
  + Mecanismos de diseño genéricos que tratan requisitos comunes como los requisitos especiales sobre persistencia, distribución, rendimiento y demás.(Diagrama de colaboración)
* **Diseño de un caso de uso**
  + Identificar las clases del diseño y/o los subsistemas cuyas instancias son necesarias para llevar a cabo el flujo de sucesos del caso de uso(Diagrama de clases)
  + Distribuir el comportamiento del CU entre los objetos del diseño que interactúan y/o entre los subsistemas participantes(Diagrama de secuencia)
  + Definir los requisitos sobre las operaciones de la clase del diseño y/o sobre los subsistemas y sus interfaces(Diagrama de clases con subsistemas)
  + Capturar los requisitos de implementación del caso de uso
* **Diseño de una clase**

El propósito de diseñar una clase es crear una clase del diseño que cumpla su papel en las realizaciones de los casos de uso y los requisitos no funcionales que se aplican a estos. Esto incluye el mantenimiento del diseño de las clases en sí mismo y los siguientes aspectos de este:

* + Sus operaciones
  + Sus atributos
  + Las relaciones en las que participa
  + Sus métodos
  + Loas estados impuestos
  + Sus dependencias con cualquier mecanismos de diseño genérico
  + Los requisitos relevantes a su implementación
  + La correcta realización de cualquier interfaz requerida
* **Diseño de un subsistema**

Los objeticos del diseño de un subsistema son:

* + Garantizar que el subsistema es tan independiente como se posible de otros subsistemas
  + Garantizar que el subsistema proporciona las interfaces correctas
  + Garantizar que el subsistema cumple su propósito de ofrecer una realización correcta de las operaciones