

UNIVERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TEMA:

**Patrones de Diseño Arquitecturales y Estructurales**

**CURSO:**

BASE DE DATOS II

**DOCENTE(ING):**

Patrick Jose Cuadros Quiroga

Integrantes:

Andree Ludwed VELASCO SUCAPUCA	(20161515)
Yofer Nain CATARI CABRERA	(2017059289)
Marko Antonio RIVAS RIOS	(2016055461)
Jorge Luis MAMANI MAQUERA	(2016055236)
Jesus ESCALANTE ALANOCA	(2015050641)
Adnner Sleyder ESPERILLA RUIZ	(2015050543)

# Índice

1. Introducción	1
2. Arquitectura de Software basada en unModelo de Capas para Entornos de CC	2
3. Patrones Arquitectónicosl	3
4. Patrones de Diseño (Facade)	4
5. Conclusiones y Trabajos Futuros	6
6. Patrones de Diseño:Especificación de la Capa de Aplicación	7

# 1. Introducción

## ¿Cómo surgen los patrones de diseño?

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño resulta ser una solución a un problema de diseño.

## 2. Arquitectura de Software basada en unModelo de Capas para Entornos de CC

.....

.....

### 3. Patrones Arquitectónicos

.....

.....

## 4. Patrones de Diseño (Facade)

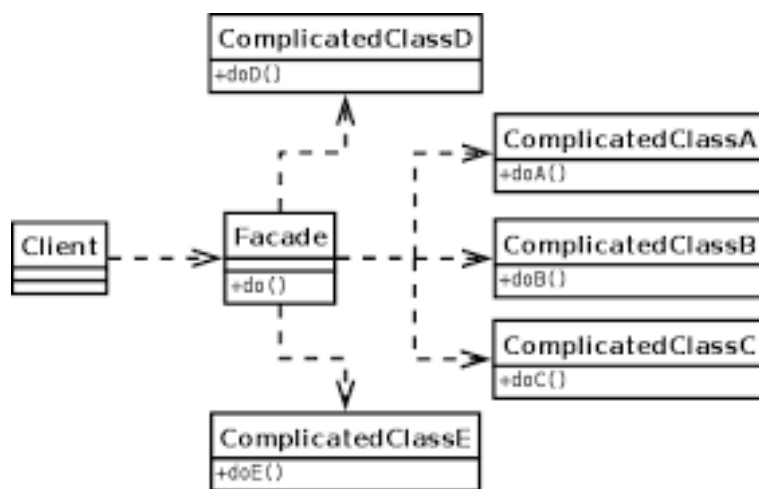
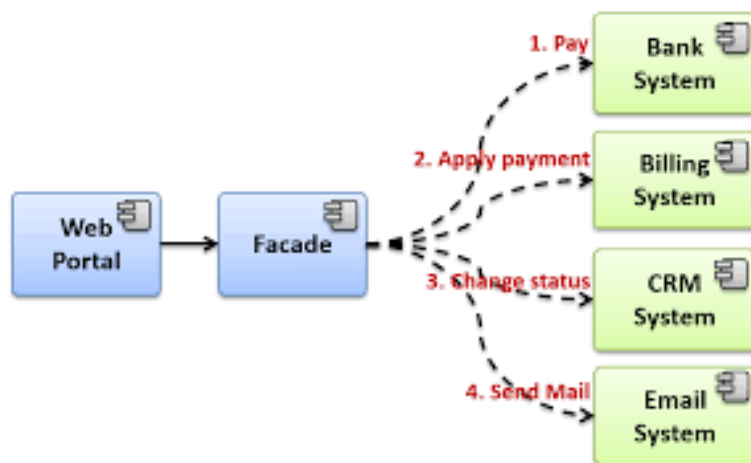
Los patrones de diseño se consideran una de las herramientas más valiosas para producir diseños de calidad y una técnica de propósito general para mejorar un diseño es identificar todas las realizaciones de patrones y aplicar reglas conocidas para mejorarlos.

- Patrones de Diseño

IDEA es un asistente de diseño interactivo para arquitectos de software destinado a automatizar la tarea de encontrar y mejorar las realizaciones de los patrones de diseño.

- ¿Qué es una fachada o facade en inglés?

Es un patrón de diseño que nos permite simplificar el interface de comunicación entre dos objetos.

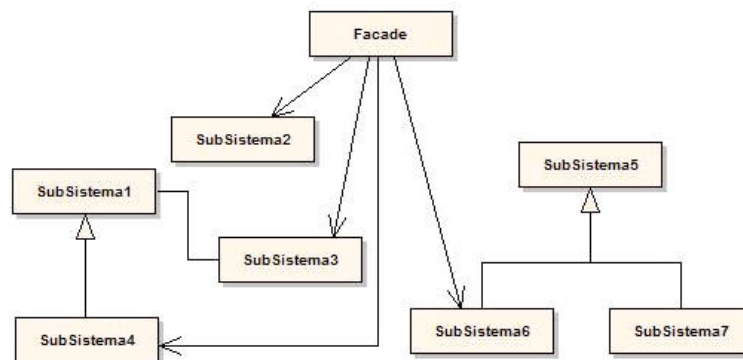


Busca simplificar el sistema, desde el punto de vista del cliente, proporcionando una interfaz unificada para un conjunto de subsistemas, definiendo una interfaz de nivel más alto. Esto hace que el sistema sea más fácil de usar.

Este patrón busca reducir al mínimo la comunicación y dependencias entre subsistemas. Para ello, utilizaremos una fachada, simplificando la complejidad al cliente. El cliente debería acceder a un subsistema a través del Facade. De esta manera, se estructura un entorno de programación más sencillo, al menos desde el punto de vista del cliente (por ello se llama "fachada").

- ¿Se debe utilizar cuando?

Se quiera proporcionar una interfaz sencilla para un subsistema complejo. Se quiera desacoplar un subsistema de sus clientes y de otros subsistemas, haciéndolo más independiente y portable. Se quiera dividir los sistemas en niveles: las fachadas serían el punto de entrada a cada nivel. Facade puede ser utilizado a nivel aplicación.



Los clientes se comunican con el subsistema a través de la facade, que reenvía las peticiones a los objetos del subsistema apropiados y puede realizar también algún trabajo de traducción. Los clientes que usan la facade no necesitan acceder directamente a los objetos del sistema.

## 5. Conclusiones y Trabajos Futuros

.....

.....



## 6. Patrones de Diseño: Especificación de la Capa de Aplicación

.....

.....