



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I TISTA GARCIA EDGAR

Asignatura: Programación Orientada a Objetos

Grupo: 03

No de Práctica(s): 13

Integrante(s): Hernández Diaz Sebastián Escamilla Suárez Magally

*No. de Equipo de
cómputo empleado:* Computador en casa

No. de Lista o Brigada: 19 y 11

Semestre: 2023-1

Fecha de entrega: 09 de Noviembre del 2022

Observaciones:

CALIFICACIÓN: _____

PRÁCTICA # 13: PATRONES DE DISEÑO

OBJETIVO: Comprender los patrones de diseño para implementar una aplicación en un lenguaje orientado a objetos utilizando alguno de ellos.

OBJETIVO DE CLASE: Distinguir entre las categorías de patrones de diseño e identificar problemas donde se pueden aplicar

ACTIVIDADES PARA EL DESARROLLO DE LA PRÁCTICA

Ejercicio 1.- Responde las siguientes preguntas basado en la guía de la práctica y en el video del día.

En caso de usar otras fuentes, indicarlas en cada pregunta)

1.- ¿Qué es un patrón de diseño y por qué es importante en la programación?

Un patrón de diseño es una solución repetible y general para problemas de ocurrencia cotidianos en el diseño de software.

Esto quiere decir que un patrón es un modelo de cómo resolver un problema que puede usarse en distintas situaciones, sería como una plantilla de qué hacer para resolver un problema pero que se pueda aplicar a distintos problemas.

Es importante para la programación porque permiten agilizar el proceso de desarrollo de solución debido a que proveen un paradigma desarrollado y probado.

Además de que la reutilización de patrones de diseño ayuda a prevenir los detalles que tal vez a simple vista no se vean pero provocan problemas más grandes, además de ayudar a la legibilidad de código para programadores ya que sería un estándar que todos podían entender y aprender.

2.- Menciona cuáles son los elementos de un patrón de diseño, y basado en la pregunta anterior, que otro elemento agregarías

Los patrones de diseño tiene 4 principales elementos:

Nombre del patrón: Este se usa para describir en grandes rasgos qué es lo que se hará en el patrón para la resolución del problema.

Problema: Este describe cuando es que se debe de aplicar el patrón, con esto se explica el problema para que sea fácil de entender además de todas sus implicaciones, ya sea formas en que se requiere la solución, propuestas para esta misma resolución y demás.

Solución: En esta parte a pesar de ser la solución no describe en específico un diseño o implementación sino que de manera abstracta provee una descripción del diseño y cómo es que se acomodan sus elementos.

Consecuencias: En este se muestran los resultados de haber aplicado el patrón, esto para evaluar cómo fue el aplicar ese patrón ya que tal vez otro patrón pudo ser

una mejor opción o sino fue lo más fácil de diseñar pero se obtuvo una reducción en el tiempo o espacio al aplicar este mismo.

Otro elemento que agregaría sería problemas de aplicar el diseño o tal vez comparaciones entre estos mismos y de esta forma dependiendo de las solicitudes se podría escoger uno u otro, también agregaría que se mostraran los problemas a la hora de implementar ya que de esta forma personas con el mismo problema que intentan resolverlo de una forma similar puedan ver cuáles han sido las fallas o problemas que el patrón trae consigo.

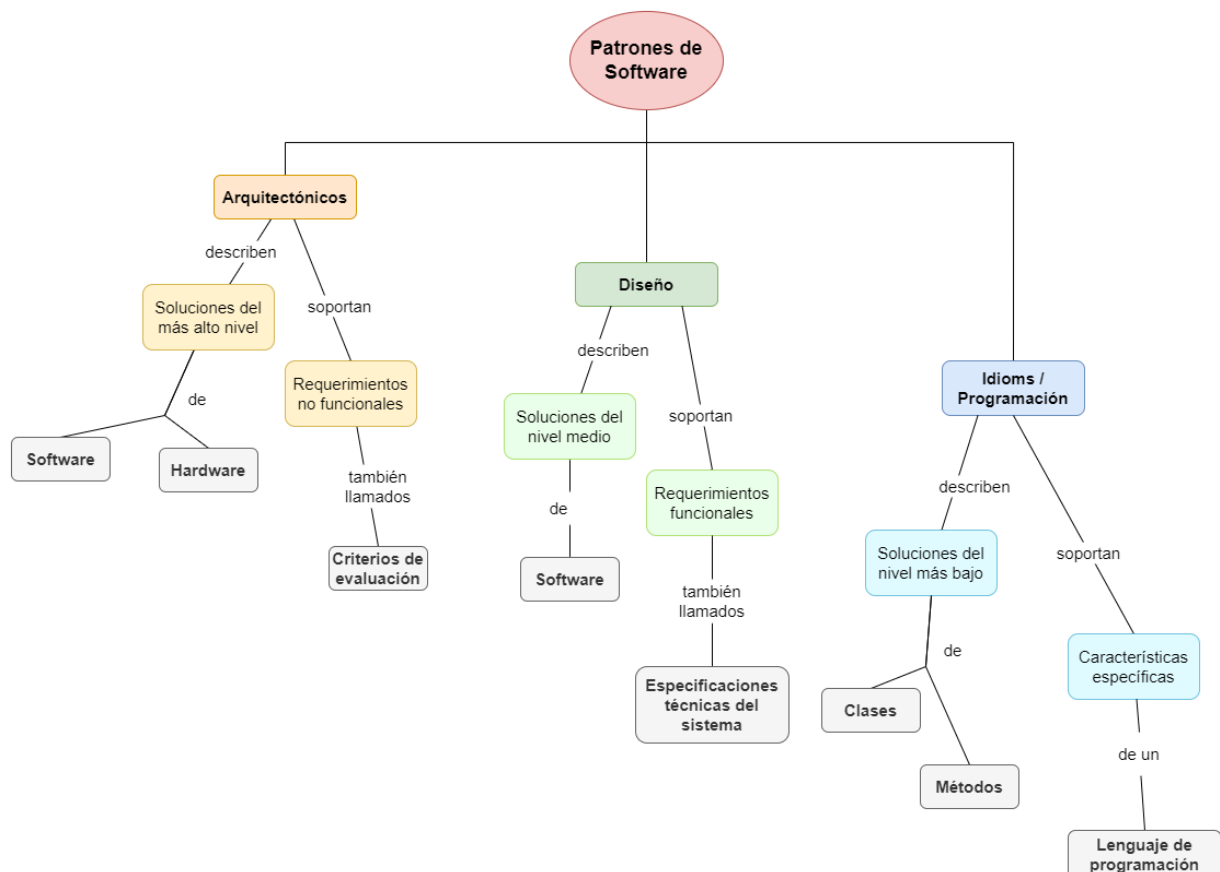
3.- ¿Cuál es la diferencia entre los patrones de diseño y UML?

Un patrón de diseño es una solución repetible y general, mientras que los diagramas UML permiten modelar aspectos conceptuales, funcionalidades y características específicas de un sistema o solución.

Mientras que se puede usar un patrón de diseño como guía para solucionar el problema en cuestión (es decir, el patrón establece una estructura base de funcionamiento), se utilizan los diagramas UML para establecer las clases, objetos, actores, y funcionamientos específicos del sistema.

Por ejemplo, el patrón de diseño te dice que para hacer un auto debe tener ruedas, motor, puertas, ventanas, etc; mientras que el UML te permite visualizar la relación entre estos y diseñar los aspectos específicos de cada parte, ver si la puerta se abre horizontal o verticalmente, si usas manivela o botón para bajar las ventanas, etc.

4.- Realiza un mapa conceptual de los tres tipos o niveles de patrones de SW



5.- Explica en tus términos en qué consiste el patrón de diseño Modelo Vista Controlador e indica a qué nivel de patrón corresponde

En este modelo se podría empezar por sus tres partes.

Modelo: Este es el que toma la responsabilidad de los datos, por ejemplo el modelo es el que se encarga de la manipulación de los mismos, es el encargado básicamente de hacer todos los procesos que se requieran pero no de mostrarlos ya que lo que hace es avisar al Controlador para que este se encargue.

Vista: Esta es como una interfaz pues es la que se encarga de reaccionar a las solicitudes del usuario y así seguir actualizándose, por ejemplo una vista sería en el inicio de sesión de una red social donde se te muestra una interfaz donde debes de agregar tu usuario y contraseña y la vista reacciona a estos para después mostrar algo más en pantalla, claro que esta parte solo es la que muestra los cambios pues al recibir la información este necesita del Controlador y del Modelo para que se puedan realizar las operaciones necesarias que este mostrara después.

Controlador: Este es un intermediario entre los elementos previamente mencionados, cumple el rol de intercambiar la información entre estos dos, podría decirse que es un puente en el cual transita la información que recibe la vista para que el Modelo haga las operaciones correspondientes y cuando este termina se mandan de la misma forma hacia la vista para que esta pueda cambiar y hacer sus respectivas operaciones también, pero no solo esto ya que también es que regula cuando un objeto ya no es necesario para que de esta forma se deshagan de él.

Por lo que en general consiste en 3 pasos, primero en la vista que es la que el usuario ve, en esta se pide información y se muestra lo visual, los cambios necesarios, después de recibir la información está para por el Controlador para que pueda llegar de manera correcta al Controlador el cual se encarga de todos los procesos necesarios para que lo demás pueda avanzar, cuando termina de hacer los cambios debe de avisar de alguna forma a la Vista y para eso avisa al Controlador de los cambios y este a su vez a la vista la cual se actualiza mediante estos cambios.

Este patrón corresponde al nivel: Es un patrón arquitectónico ya que consiste en las interrelaciones que hay entre los 3 elementos mencionados con anterioridad y como estos se comportan con las reacciones que recibe cada uno por parte del usuario.

6.- Explica dos ejemplos de problemas en los cuales se podría utilizar el patrón Singleton y por qué es adecuado en ellos

Un ejemplo de problema sería al ingresar a una carrera, como estudiante pues sólo se permite ingresar una vez por persona, esto haría que en el sistema solo aparezca una vez tu nombre con tus características que serían la edad, carrera y además, sería adecuado usar el patrón Singleton porque solo existiría esa instancia del

alumno, no se podría crear otra que tenga el mismo nombre y sus mismas características.

Otro ejemplo seria en trámites de gobierno por ejemplo el servicio militar donde solo se puede ingresar una vez y solo una persona, por lo que tus datos deberían de quedar registrados una sola vez y de esta forma no se podría crear otra persona con tus mismos datos tratando de cumplir ese trámite ya que al hacerlo una vez es suficiente, por eso el patrón Singleton es bueno ya que no se requiere hacer ese trámite más de una vez por lo cual solo se crearía una instancia y con esa sería suficiente.

7.- Explica en tus términos en qué consiste el patrón Observer y menciona un ejemplo de aplicación donde se pueda aplicar

Este patrón es el encargado de controlar la dependencia de los objetos, este mismo las asigna y en base a esas asignaciones es como se trabaja, también es el mantiene la comunicación entre los objetos pues informa sobre los cambios que se llevan a cabo para que los demás objetos que sean dependientes de este puedan hacer cambios de manera correcta y adecuada al funcionamiento que se requiere. Podría decirse que es la interfaz con la que se trabaja pues es el medio de comunicación entre el usuario y el código.

Un ejemplo de la utilización de este patrón podría ser por ejemplo en youtube en donde si estás suscrito a un canal no debes de siempre estar verificando si hay un nuevo video ya que el mismo sistema se encargará de avisarte cuando esto ocurra, en cambio si no estas suscrito no hay una dependencia en los datos sobre tu cuenta y el canal por lo cual no serás avisado y debería de revisar manualmente si es que hay un nuevo video.

8.- Explica cuales son las diferencias entre los patrones Observer y Composite

Las diferencias entre estos dos patrones son que el patrón composite solo funciona cuando los problemas se pueden modelar en forma de un árbol, mientras que en el patrón Observer esto no ocurre y hay más flexibilidad al momento de crear estas dependencias.

La ventaja de esta Composite es que no tienes que preocuparte por las clases concretas de los objetos que componen el árbol.

Como se mencionó el patrón composite solo permite construir objetos complejos partiendo de otros más sencillos utilizando una estrategia de composición recursiva, por otro lado el patrón observer no es necesario esto ya que tiene un alto grado de independencia lo cual es contrario al composite ya que este si es dependiente.

9.- Escribe las conclusiones de la práctica de manera individual

Escamilla Suárez Magally:

Esta práctica, aunque sencilla en el sentido de que fue sólo un cuestionario, nos ayudó a comprender qué son los patrones de diseño, distinguir entre sus categorías y distintos patrones existentes.

El hecho de que se hiciera el mapa conceptual sobre los patrones de software me ayudó no solo a comprender las diferencias entre los 3 tipos (entre niveles y requerimientos de cada uno), si no que además me permitió visualizar la “jerarquía” que se presenta entre ambos. Antes pensaba que se tenía que escoger uno y ya para generar soluciones de sw; ahora veo que los 3 funcionan de manera integral para diseñar todas las partes del sistema.

Otra cosa que hay que destacar de esta práctica fue el hecho de que varias preguntas pedían aplicaciones de ciertos patrones. Esto es muy importante, ya que entender la teoría no es suficiente para la programación, y el ponerse a pensar “¿En dónde me sirve este patrón?” hace que verdaderamente se comprendan las diferencias entre los patrones, el enfoque que cada uno tiene, y aplicaciones de la vida real, que muy posiblemente en nuestra carrera laboral tendremos que implementar.

Me gustaría poder profundizar más en los patrones de diseño, pero entiendo que son un tema complejo y que necesitaríamos meses para poder aprender bien sobre ellos. Por lo que esta práctica estuvo bastante bien para conocer lo básico.

Hernandez Diaz Sebastian:

En esta práctica se cumplieron los objetivos de manera correcta puesto que se revisaron los distintos tipos de patrones de diseño, también se revisó su clasificación y a pesar de que en cada uno de ellos la explicación era algo sencilla se pudo entender el concepto y como es que estos mismos se aplican en los problemas a resolver.

En la resolución de la práctica a pesar de ser un cuestionario fue algo difícil pues eran muchos patrones a entender y saber cómo aplicar estos en los problemas que surgen, aunque esto también es una ventaja pues nos muestra cómo los problemas que nos pasan a nosotros también le han pasado a programadores en el pasado y estos los han resuelto de una manera, dejando sus conocimientos para que las personas puedan ayudarse en ellos y resolverlos.

Los patrones de diseño por eso son tan importantes pues nos proveen de una forma de solucionar nuestros problemas por los cuales muchas personas han pasado pero

de una forma más general pues no tendra la solucion para nuestro problema en especifico, pero nos ayuda mucho en saber cómo separar el problema, poner todo de una manera en la cual todo es más ordenado y así llegar a una resolución con la cual estemos satisfechos y sea una que tambien nos facilito el trabajo.

Es un tema en el cual para entenderlo de manera correcta y llegar a dominarlo se necesita de mucho esfuerzo e investigación ya que como mencione solo vimos una pequeña explicación sobre el concepto de estos mismos pero en las preguntas donde se nos pedía ejemplos sobre estos era lo interesante pues nos ponía a prueba para entender de mejor manera esos patrones.

Es por eso que esta práctica aunque corta fue una que nos enseña un tema muy grande y complejo en el cual debemos de investigar más pues es un tema muy importante en la resolución de estos problemas de programación.