



Instituto Tecnológico de Oaxaca



Asignatura: Patrones de diseño y componentes.

Profesor: Limón Cordero Rogelio Noé.



Hora: 11:00 - 12:00

Grupo: ISA

Alumnos:

César Santiago Guzmán.

Omar Edwíng Vásquez Alonso

Carrera: Ingeniería en Sistemas Computacionales.

Lugar y fecha: Oaxaca de Juárez a 25 de Abril de 2017.

Problemática:

En una biblioteca de publica no se obtiene un registro eficaz sobre las consultas de libros y otros materiales ya que no se guardan dichos registros y es difícil obtener datos estadísticos de los mismos por la cuestión de leer las fichas de registros de todos los materiales que disponen la biblioteca, así mismo no saber qué cantidad de determinados libros se cuentan a disposiciones de los diferentes visitantes a la biblioteca. Además de la confusión de se tiene cuando los visitantes desean llevarse un recurso de la biblioteca y no cuentan con credencial y desconocen los beneficios de ser socios de la misma.

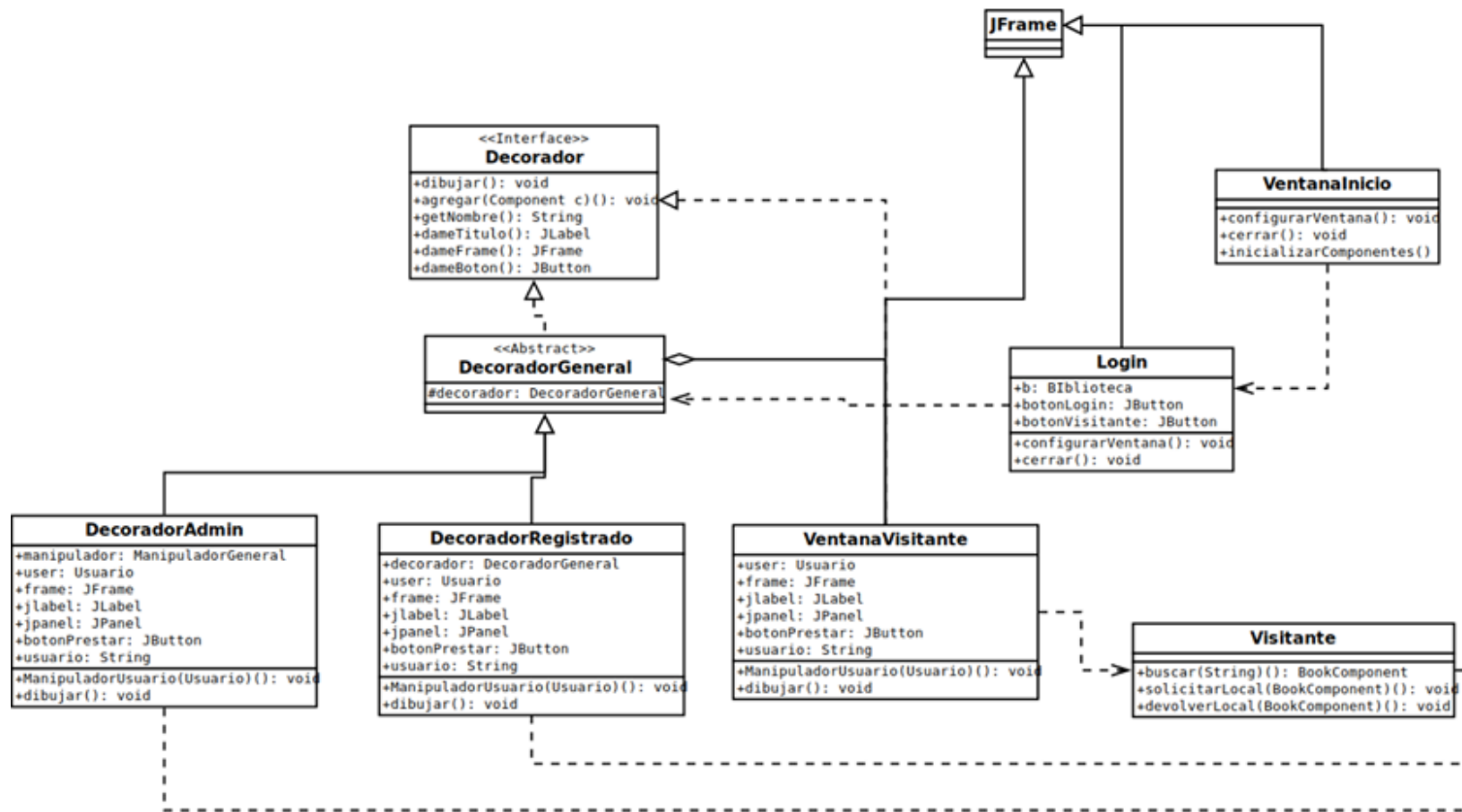
Solución:

Para poder arreglar este tipo de problemáticas es necesario implementar un grupo de patrones de diseños en combinación con interfaces graficas de usuario, con las cuales se puede llevar un mejor control de los patrones y las operaciones sobre la base de datos, los patrones a usar son:

- **Patrón Singleton:** Solucionara el problema de que muchos usuarios quieran acceder a una misma instancia en especial a la instancia Administrador.
- **Patrón Iterador:** En las búsquedas se recorrerán los diferente libros y materiales
- **Patrón Manipulador:** Las interfaces y funciones accesibles de los usuarios serán diferentes de acuerdo con los roles y permisos de cada uno.
- **Patrón Estrategia:** Comportamientos como el préstamo de materiales serán distintos dependiendo de dicho material.
- **Patrón Observador:** Cuando haya cambios en los estados de los libros estos desencadenaran observadores que actúen dependiendo de la situación.
- **Patrón Composite:** Algunas operaciones se realizaran sobre libros o colecciones de libros por igual
- **Patrón Decorador:** Para solventar el uso de ventanas dependiendo del rol que tengan los usuarios y funciones de estos.

Ventajas de usar patrones sobre no usarlos.

- **Patrón Singleton:** Al no usar este patrón se corre el riesgo de abrir dos instancias de la clase Administrador así evitando el conflicto de uso de operaciones al mismo tiempo.
- **Patrón Iterador:** Cada una de los elementos se guardan en diferentes tipos de estructuras, con este patrón se identifica y se recorren cada una de ellas sin problemas.
- **Patrón Manipulador:** El patrón desarrolla por cada rol un diferente uso de las funciones y la manipulación de los datos y características de la interfaz gráfica.
- **Patrón Estrategia:** Este patrón identifica los tipos de elementos para que no utilice funciones iguales en diferentes elementos.
- **Patrón Observador:** Cuando haya cambios en los estados de los libros estos desencadenaran observadores que actúen dependiendo de la situación.
- **Patrón Composite:** Algunos artículos se realizan sobre tipo de libros, revistas y otros artículos por igual y al no usar este patrón se duplica la redundancia de código por cada uno de los artículos.
- **Patrón Decorador:** Para no crear una ventana por cada usuarios ya que estos comparten características similares.



2

3

