Separación de responsabilidades en el proyecto

1. Capa de servicios

Encapsula lógica de negocio y operaciones relacionadas con los datos. Los servicios interactúan con la base de datos o cualquier otra fuente de datos:

* CRUD:
  + Crear, editar, borrar y obtener registros en la base de datos
* Validar datos previa interacción con la DB
* Reglas de negocio. implementar lógica específica.

1. Capa de controladores

Actúa como intermediario entre la vista (UI) y los servicios. Manejan las solicitudes de la vista y delegan la lógica de negocio a los servicios:

* Delegación de operaciones a los servicios
* Transformación de datos
* Manejo de errores

1. Capa de managers

Gestiona recursos técnicos específicos, como formularios o tablas. Los manager se encargan del ciclo de vida, la configuración y la interacción técnica de estos recursos.

* Gestión de formularios
  + Registra formularios activos
  + Abre formularios basados en configuraciones predefinidas.
* Centralización de lógica técnica:
  + Manejas botones comunes
  + Validación básica
  + Cierre de formularios.
* Optimización:
  + Evitar duplicación de instancias
  + Gestión de ciclo de vida de los recursos

1. Capa de configuración

Centraliza configuraciones comunes para formularios, tablas y otros componentes. Define cómo se comportan y se presentan los recursos.

* Definición de configuraciones
* Centralización

1. Capa de vista

Presenta datos al usuario y maneja la interacción con la interfaz gráfica. La vista no debe contener ninguna lógica de negocio.

* Presentación de datos
* Interacción
  + Clicks de usuario o selección de elementos.
* Delegación:
  + Llamar a métodos de controladores para manejar eventos.

Relación entre capas

* Vista: interactúa con el usuario y delega eventos al controlador
* Controlador: procesa las solicitudes de la vista y delega la lógica de negocio al servicio
* Servicio: maneja la lógica de negocio y las operaciones con la base de datos.
* Manager: recursos técnicos como formularios y tablas.
* Configuración: define cómo se comportan y presentan los recursos.

CASO DE USO TIPO

* Un usuario selecciona un elemento en la tabla de requisitos que está en la sección (stacked) de requisitos en la página principal (home).
* Los botones de editar y borrar se activan cuando ocurre la selección.
* El usuario clicka en editar y se abre un formulario con los datos de dicha línea cargados en el formulario.
* El usuario cambia un par de datos y pulsa en aceptar.
* El sistema revisa el nuevo input y actualiza la base de datos con los nuevos datos introducidos.
* La tabla se actualiza y muestra al usuario todos los elementos. La selección deja de existir y los botones vuelven a desactivarse.

Relación entre capas para dicho caso de uso:

* Vista (UI):
  + Mostrar la tabla de requisitos al usuario.
  + Detectar la selección de un elemento en la tabla.
  + Activar/desactivar botones según el estado de la selección.
  + Abrir el formulario de edición con los datos cargados.
  + Mostrar mensajes de error o éxito al usuario.
  + Interacciones:
    - Con controller para desplegar la lógica de negocio.
    - Recibe datos del controller para actualizar la tabla
* Controlador:
  + Maneja eventos de la vista (clickar en editar)
  + Delega la lógica de negocio al servicio
  + Transforma los datos entre la vista y el servicio (convierte datos de formulario en formato adecuado para el servicio)
  + Manejar errores del servicio y devolver mensajes claros a la vista.
  + Interacciones:
    - Con la vista para recibir eventos y actualizar la interfaz
    - Con el servicio para realizar operaciones de negocio
* Servicio:
  + Valida los datos recibidos del controlador.
  + Interactúa con la base de datos para realizar operaciones (actualizar un requisito, pe)
  + Implementa reglas de negocio específicas (verificar que los datos sean válidos)
  + Interacciones:
    - Con el controlador para recibir datos y devolver resultados
    - Con la base de datos para realizar operaciones CRUD
* Manager:
  + Gestiona el ciclo de vida del formulario (abrirlo, cerrarlo, registrar formularios activos)
  + Maneja la lógica técnica del formulario (conectar botones, manejar validación básica)
  + Interacciones:
    - Con la vista, para abrir y cerrar formularios.
    - Con el servicio para delegar operaciones de negocio.
* Base de datos:
  + Almacena y recupera datos
  + Provee interfaz para operaciones CRUD
  + Interacciones:
    - Con el servicio, para recibir solicitud de operaciones

Flujo del caso de uso

1. Vista:

* El usuario selecciona un elemento de la tabla de requisitos.
* La vista detecta la selección y activa los botones “Editar” y “Eliminar”

1. Controlador:

* La vista llama al controlador para manejar el evento de click en “Editar”.
* El controlador solicita al servicio los datos del requisito seleccionado.

1. Servicio:

* El servicio interactúa con la base de datos para obtener los datos del requisito.
* Devuelve los datos al controlador.

1. Controlador:

* El controlador pasa los datos al mánager para abrir el formulario de edición.

1. Manager:

* El manager abre el formulario y carga los datos en los campos correspondientes.

1. Vista:

* El usuario edita los datos y hace clic en “Aceptar”.
* La vista llama al controlador para manejar el evento de clic en “Aceptar”.

1. Controlador:

* El controlador valida los datos del formulario utilizando el servicio.
* Si los datos son válidos, el controlador solicita al servicio que actualice el requisito en la base de datos.

1. Servicio:

* El servicio valida los datos y actualiza el requisito en la base de datos.
* Devuelve un resultado al controlador.

1. Controlador:

* El controlador solicita al servicio los datos actualizados de la tabla.
* Devuelve los datos a la vista.

1. Vista:

* La vista actualiza la tabla con los datos actualizados.
* La selección se elimina y los botones “Editar” y “Eliminar” se desactivan.