Inicio del proyecto



Z C2.4 Estilos, patrones de arquitectura de software

Mapa mental sobre los estilos y patrones de arquitectura



Instrucciones

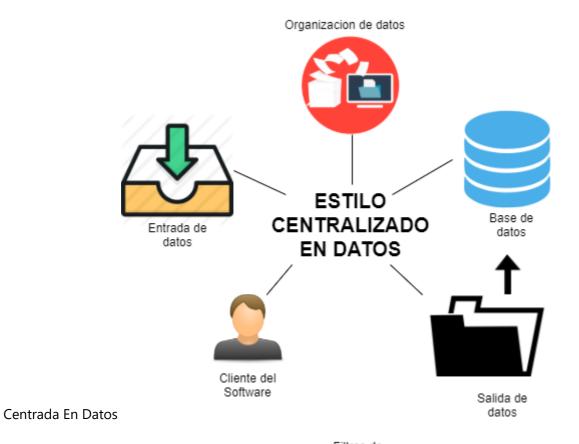
- De acuerdo con la documentación indicada por el asesor sobre el tema estilos, y patrones de arquitectura elaborar un mapa mental.
- Toda actividad o reto se deberá realizar utilizando el estilo MarkDown con extension .md y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento single page, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces, y debe ser nombrado con la nomenclatura C2.x_NombreApellido_Equipo.pdf.
- Es requisito que el .MD contenga una etiqueta del enlace al repositorio de su documento en GITHUB, por ejemplo Enlace a mi GitHub y al concluir el reto se deberá subir a github.
- Desde el archivo .md exporte un archivo .pdf que deberá subirse a classroom dentro de su apartado correspondiente, sirviendo como evidencia de su entrega, ya que siendo la plataforma oficial aquí se recibirá la calificación de su actividad.
- Considerando que el archivo .PDF, el cual fue obtenido desde archivo .MD, ambos deben ser idénticos.
- Su repositorio ademas de que debe contar con un archivo **readme**.md dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, evite utilizar texto para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

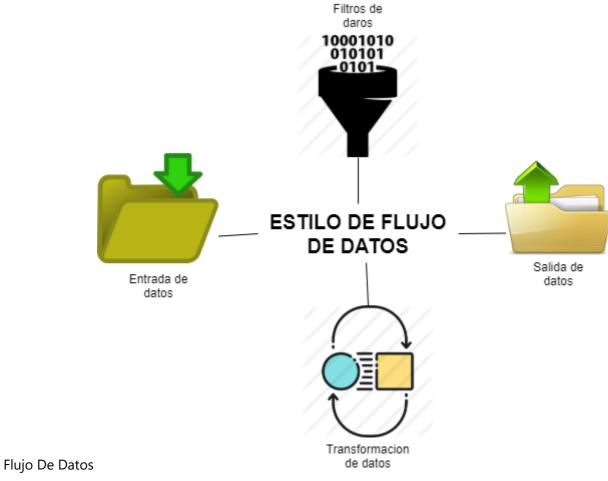
```
readme.md
blog
 | C2.1 x.md
  | C2.2 x.md
  C2.3 x.md
 C2.4_x.md
 img
docs
| A2.1_x.md
| A2.2_x.md
```



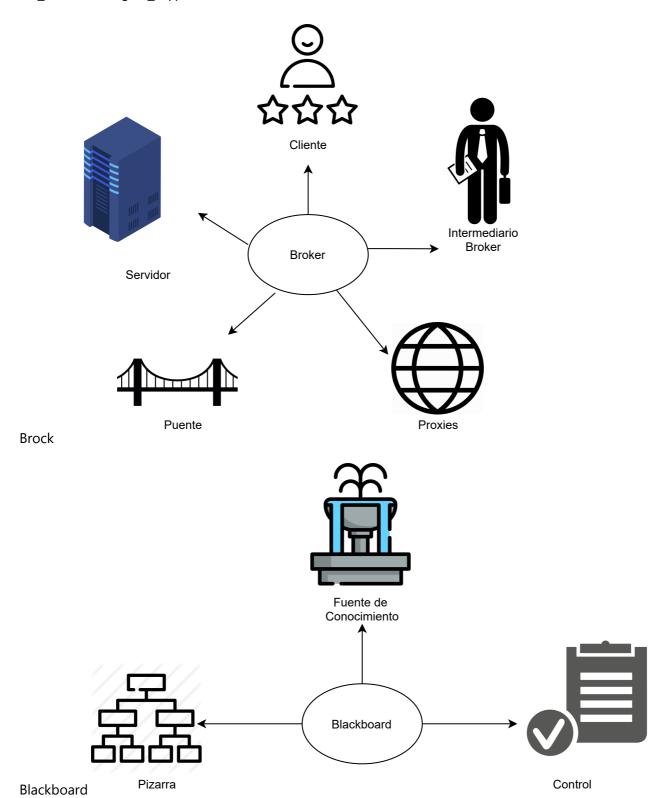
Nesarrollo 🍑

1. Mapa mental sobre los estilos, y patrones de arquitectura de software.

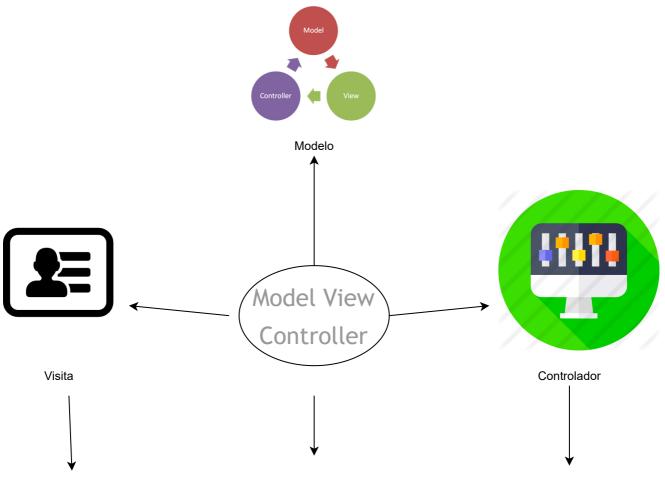








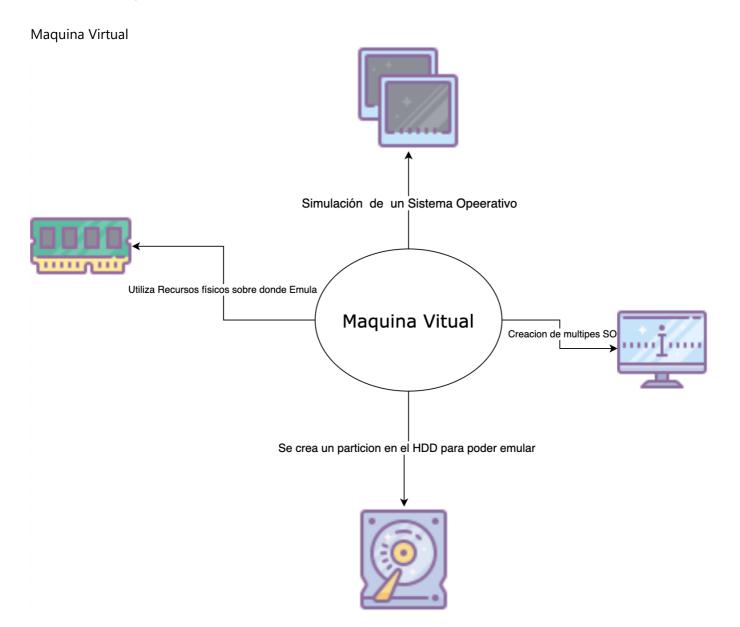
Model View Controller



Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto requiere de dicho 'modelo' la información que debe representar como salida.

Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio)

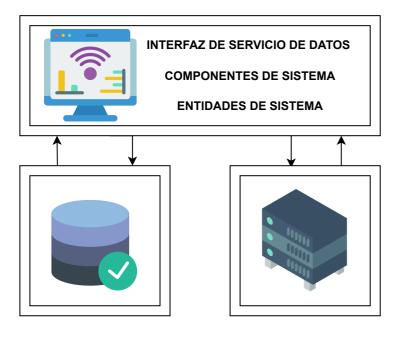
Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos)







CAPA DE PRESENTACION



LOGICA DE NEGOCIO

ORIGENES DE DATOS

SERVICIOS



Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	20
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	80

🕮 Ir a mi github