

PARCIAL 1

ARQUITECTURA DE SISTEMAS DE INFORMACION

UNIVERSIDAD SANTIAGO DE CALI

LIBIA MARCELA CHAVARRO ORTIZ

15 SEPTIEMBRE 2020

1. La arquitectura de software son las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. Los elementos pueden ser entidades que existen en tiempo de ejecución (objetos, hilos), entidades lógicas que existen en tiempo de desarrollo (clases, componentes) y entidades físicas (nodos, directorios). Por otro lado, las relaciones entre elementos dependen de propiedades visibles (o públicas) de los elementos, quedando ocultos los detalles de implementación. Finalmente, cada conjunto de elementos relacionados de un tipo particular corresponde a una estructura distinta, de ahí que la arquitectura está compuesta por distintas estructuras.

En informática, los sistemas de información ayudan a administrar, recolectar, recuperar, procesar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada organización.

La importancia de un sistema de información en la arquitectura de software radica en la eficiencia en la correlación de una gran cantidad de datos ingresados a través de procesos diseñados para cada área con el objetivo de producir información válida para la posterior toma de decisiones.

2. Audio adjunto
3. Audio adjunto
4. Ventajas:
  - Pues como nuestra aplicación se trata de llevar un control sobre las finanzas del usuario, la API Rest ayudada a que el producto se puede escalar con un equipo de desarrollo sin que ello represente muchas dificultades.
  - Al final solo te tienes que preocupar que el nexo cliente / servidor esté correcto. Puedes hacer cambios en tu servidor, lenguajes, bases de datos, etc. y mientras devuelvas los datos que toca todo irá correctamente sin que afecte a los datos del usuario
  - No mantener el estado, no requiere memoria, se pueden atender más peticiones
  - No requiere escribir el HTML, por lo tanto, tienes menos procesamiento en el servidor

Desventajas:

- Puede producirse en determinadas circunstancias mayor rigidez en el desarrollo, sobre todo al ser dos proyectos independientes, tu back basada en REST y el/los frontales, pueden surgir situaciones de des-sincronización. Por ejemplo, desde el cliente detectas que necesitas nuevas cosas del API y los encargados de crearlas en la parte del server pueden estar a otro ritmo y tardar en desarrollarla

5.

6. Audio adjunto

7.

<b>Arquitectura</b>	<b>¿Que es?</b>	<b>Diferencias</b>
<i>MVC</i>	un patron arquitectural, un modelo o guia que expresa como organizar y estructuras los componentes de un sistemas de software.Un patrón de arquitectura expresa un esquema de organización estructural fundamental para sistemas de software. Proporciona un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y directrices para organizar las relaciones entre ellos	en este patron de diseño se separan los datos de una aplicación, la interfaz de usuario, y la logica de negocio en tres componentes distintos.
<i>MVP</i>		El modelo de una interfaz que define los datos que se mostrara o no actuado en la interfaz de usuario. _el presentador actua sobre el metodo y la vista. _la vista es una interfaz pasiva que exhibe datos(modelo) y ordenes de usuario de las rutas (eventos) y los formatea para mostrarles vista
<i>MVVM</i>		en este patron de diseño se separan los datos de una aplicación, la interfaz de usuario pero en vez de controla manualmente los cambios en la vista o en los datos, esto se actualizan directamente cuando sucede un cambio en ellos
<i>ADR</i>		_El dominio solo recibe informacion de la accion, no del respondedor. _el control de ejecucion pasa al respondedor despues de que la accion termina de interactuar con el dominio , es decir, el respondedor es completamente responsable de generar todos los resultados