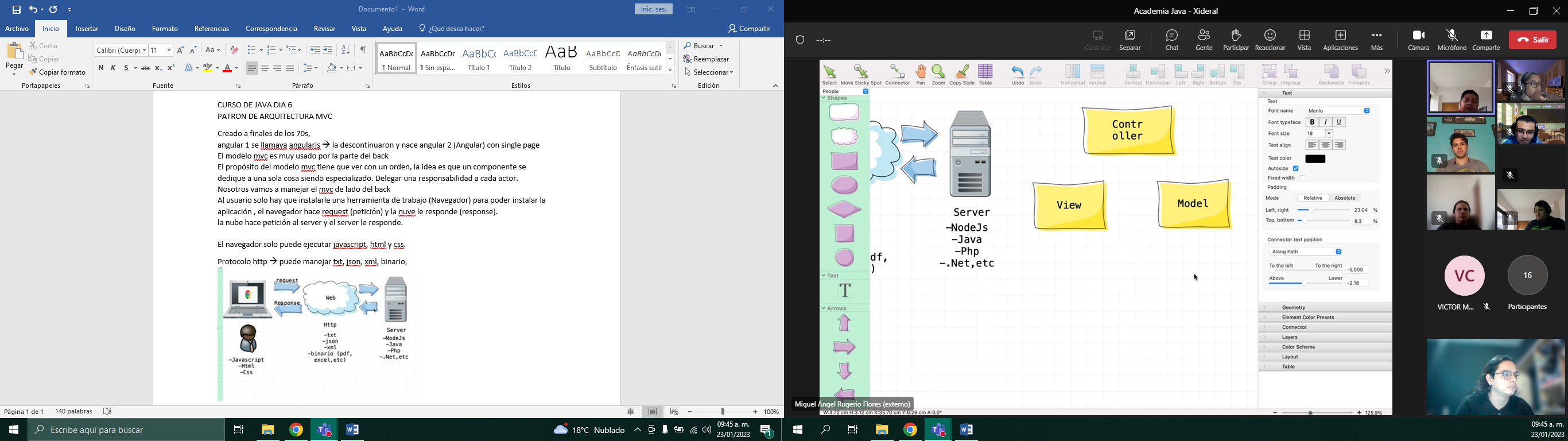
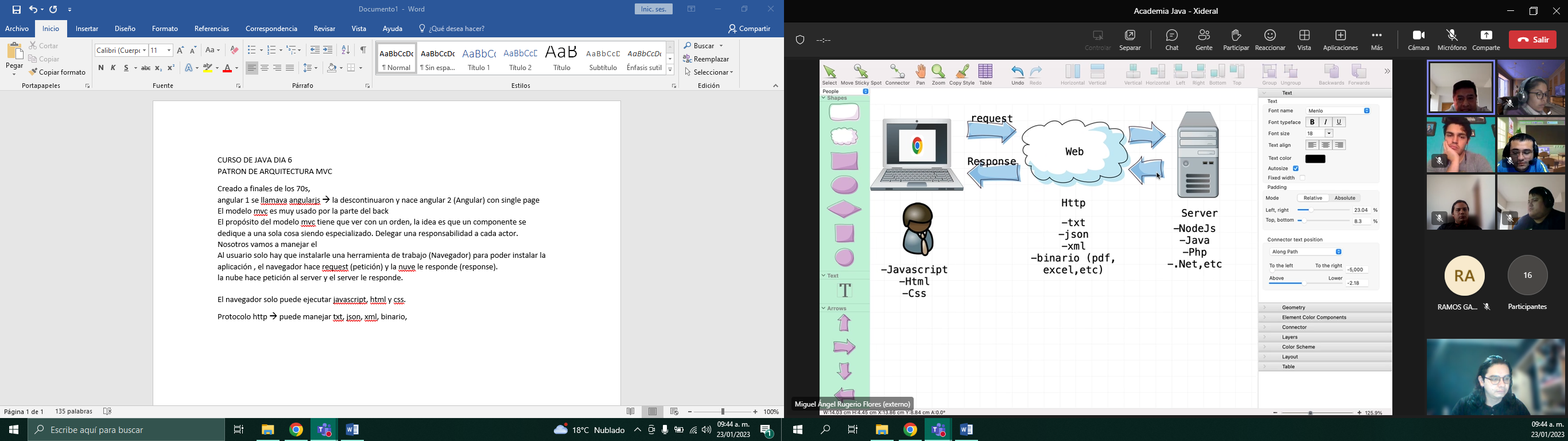
CURSO DE JAVA DIA 6  
PATRON DE ARQUITECTURA MVC

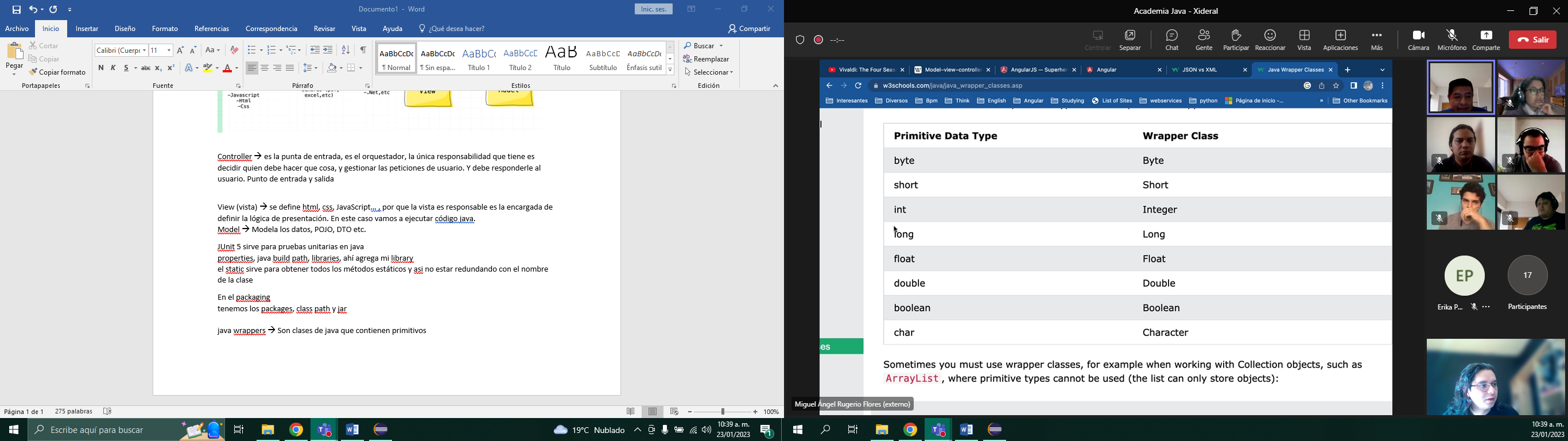
Creado a finales de los 70s,   
angular 1 se llamava angularjs 🡪 la descontinuaron y nace angular 2 (Angular) con single page  
El modelo mvc es muy usado por la parte del back  
El propósito del modelo mvc tiene que ver con un orden, la idea es que un componente se dedique a una sola cosa siendo especializado. Delegar una responsabilidad a cada actor.  
Nosotros vamos a manejar el mvc de lado del back  
Al usuario solo hay que instalarle una herramienta de trabajo (Navegador) para poder instalar la aplicación , el navegador hace request (petición) y la nuve le responde (response).   
la nube hace petición al server y el server le responde.  
  
El navegador solo puede ejecutar javascript, html y css.

Protocolo http 🡪 puede manejar txt, json, xml, binario,   


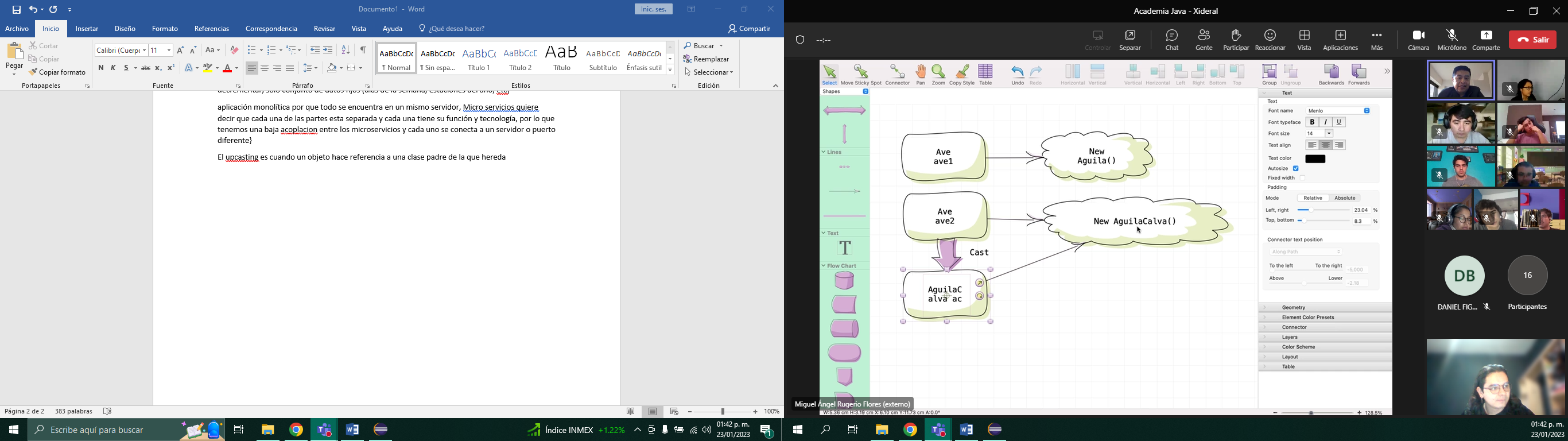
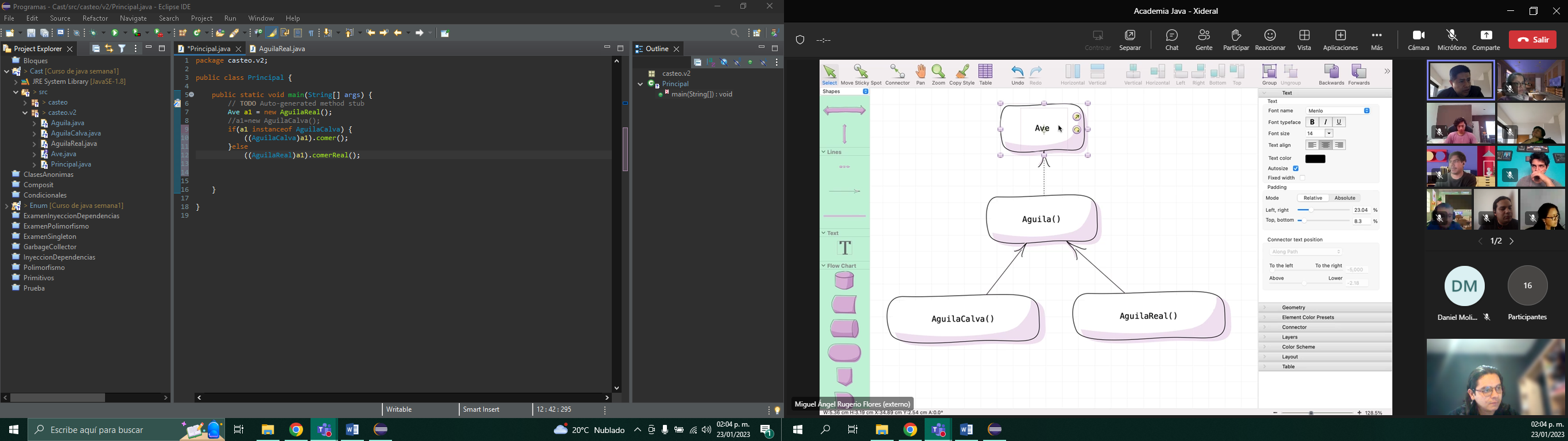
Controller 🡪 es la punta de entrada, es el orquestador, la única responsabilidad que tiene es decidir quien debe hacer que cosa, y gestionar las peticiones de usuario. Y debe responderle al usuario. Punto de entrada y salida

View (vista) 🡪 se define html, css, JavaScript… , por que la vista es responsable es la encargada de definir la lógica de presentación. En este caso vamos a ejecutar código java.  
Model 🡪 Modela los datos, POJO, DTO etc.

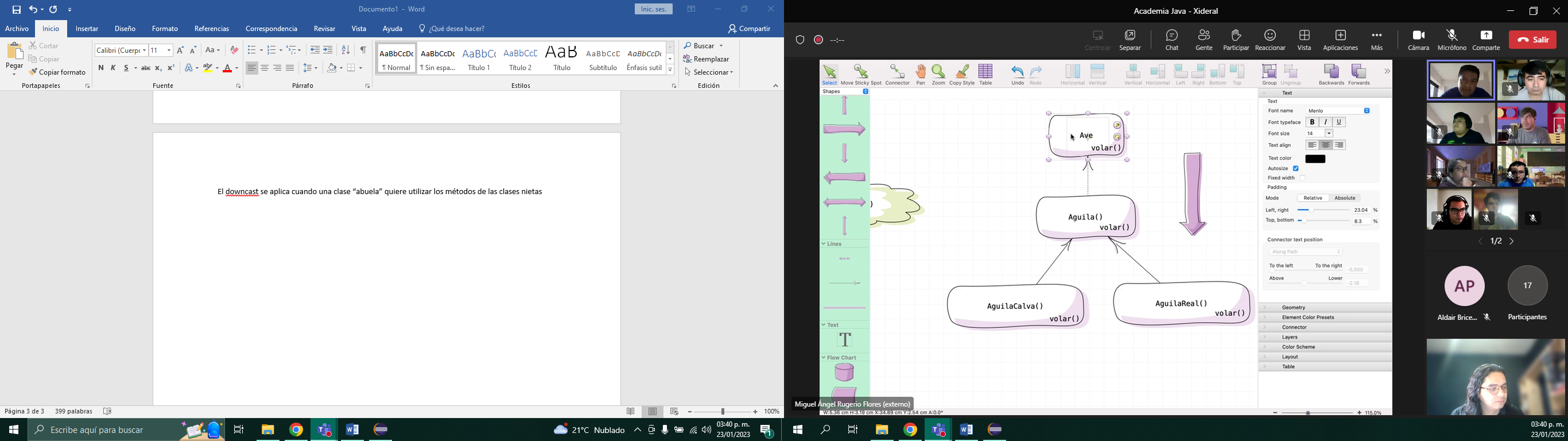
JUnit 5 sirve para pruebas unitarias en java  
properties, java build path, libraries, ahí agrega mi library  
el static sirve para obtener todos los métodos estáticos y asi no estar redundando con el nombre de la clase

En el packaging   
tenemos los packages, class path y jar  
  
java wrappers 🡪 Son clases de java que contienen primitivos  
  
Cuando usar wrapper en lugar de primitivos? Depende del contexto.  
El enum es un conjunto de datos que permanecen constantes, es decir, no deberían aumentar o decrementar, solo conjunto de datos fijos (días de la semana, estaciones del año, etc)

aplicación monolítica por que todo se encuentra en un mismo servidor, Micro servicios quiere decir que cada una de las partes esta separada y cada una tiene su función y tecnología, por lo que tenemos una baja acoplacion entre los microservicios y cada uno se conecta a un servidor o puerto diferente}

El upcasting es cuando un objeto hace referencia a una clase padre de la que hereda  
  


El downcast se aplica cuando una clase “abuela” quiere utilizar los métodos de las clases nietas



Los comportamientos los tomamos de los objetos, los atributos los tomamos de la variable

Si yo tengo un método que regresa una clase padre, puedo en su lugar, regresar una clase hija, eso es el covariant

GENERICS  
El comparable solamente puedo comparar por un criterio