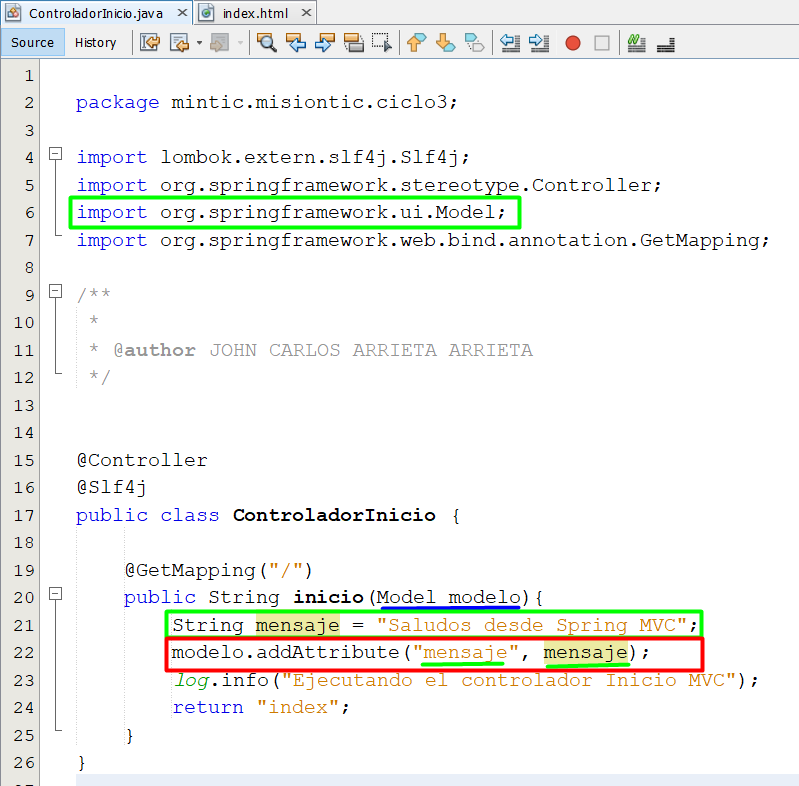
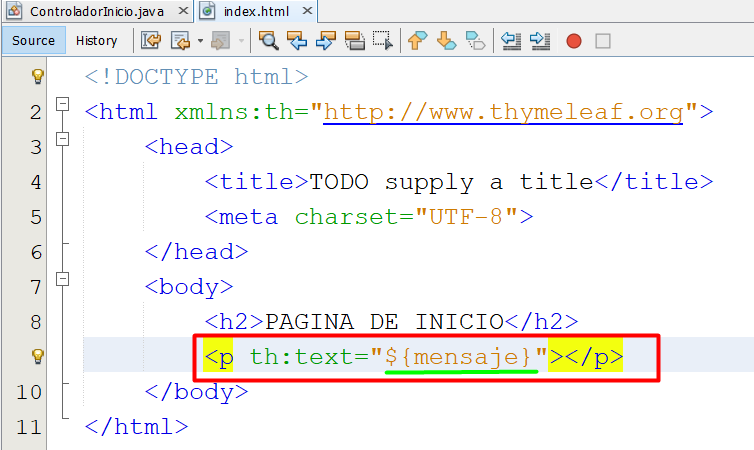
Ahora aprenderemos a enviar datos desde el controlador a la vista, de tal manera que estos datos se puedan Presentar dinámicamente en la página.

Para lograrlo usaremos un objeto de la clase Model, clase que ya está.Incorporada en el API Spring.

Los objetos de esta clase actúan como contenedores de objetos de tipo diccionario o Map, es decir Pareja de clave valor.

Posteriormente, modificaremos la vista, es decir, nuestro archivo index.html Para recuperar la clave y el valor enviados por el controlador y poder visualizar o manipular el dato enviado.

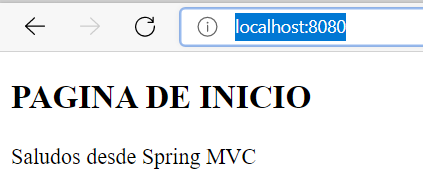


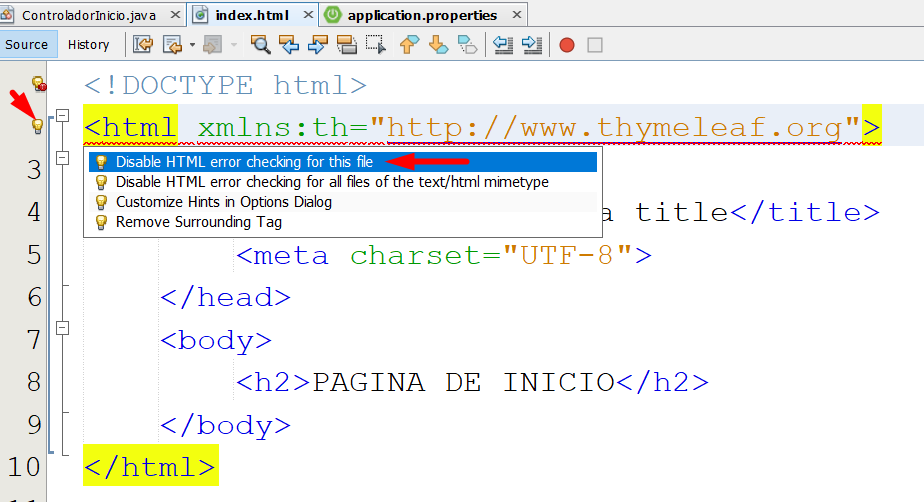
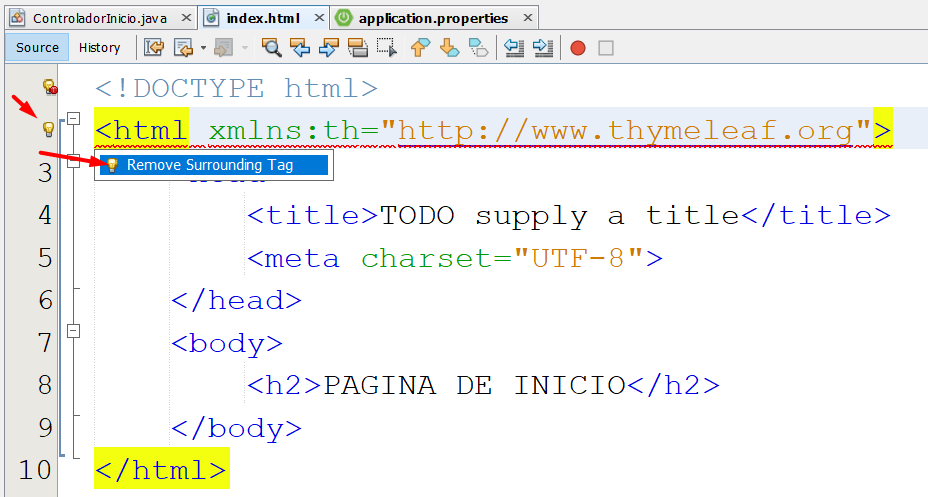


Guardamos los cambios.

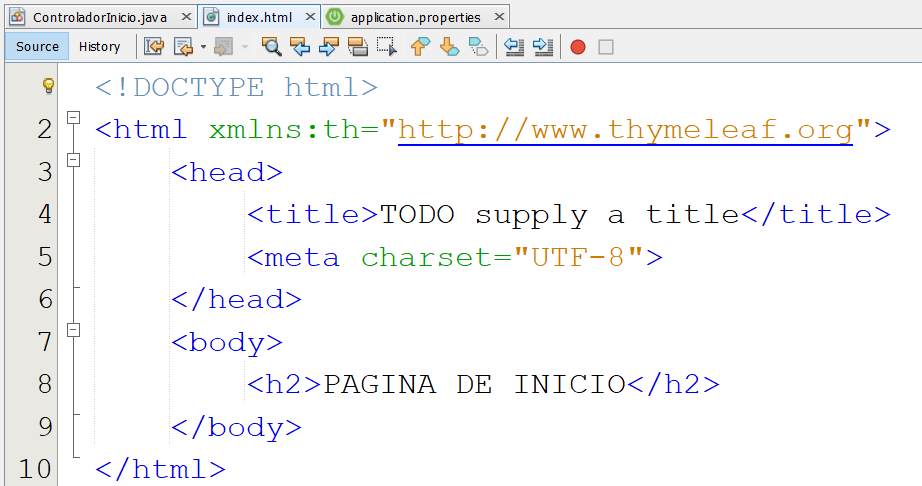
Ejecutamos la aplicación.

Abrimos La URL <http://localhost:8080/> en el navegador Para que realice una petición al controlador Solicitando la ruta raíz, la cual mapea el método inicio el cual realiza la operación final.

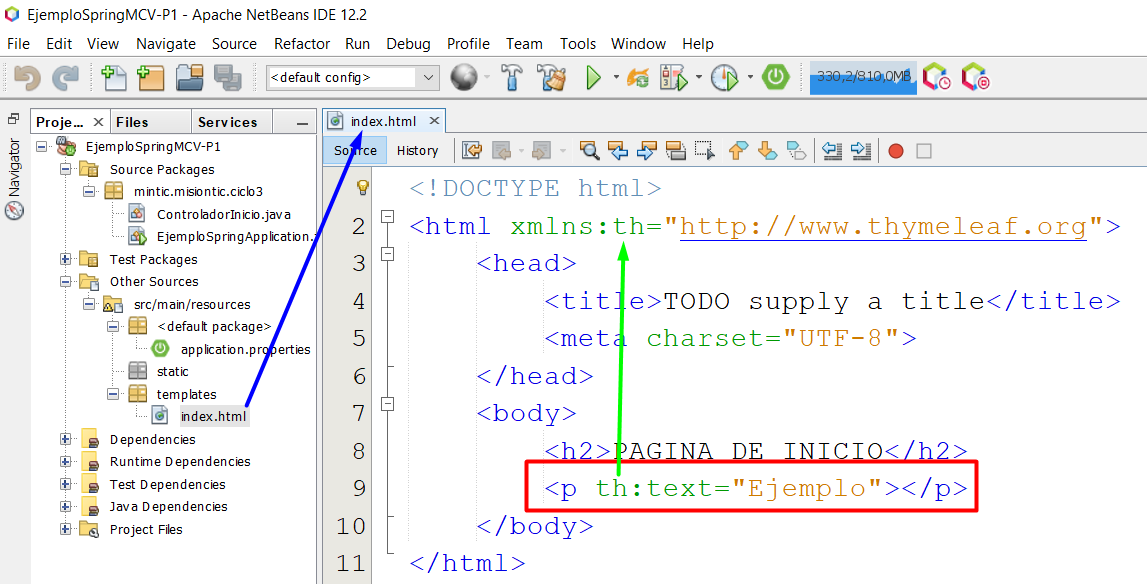




La forma de inyectar variables a nuestro html con el motor de plantillas thymeleaf es con el atributo “th:”. Thymeleaf, nos da varias opciones para introducir distintos tipos de variables, en este caso es un texto.



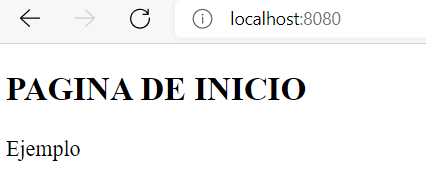
agregamos un atributo a la etiqueta de párrafo



Guardamos

ejecutamos

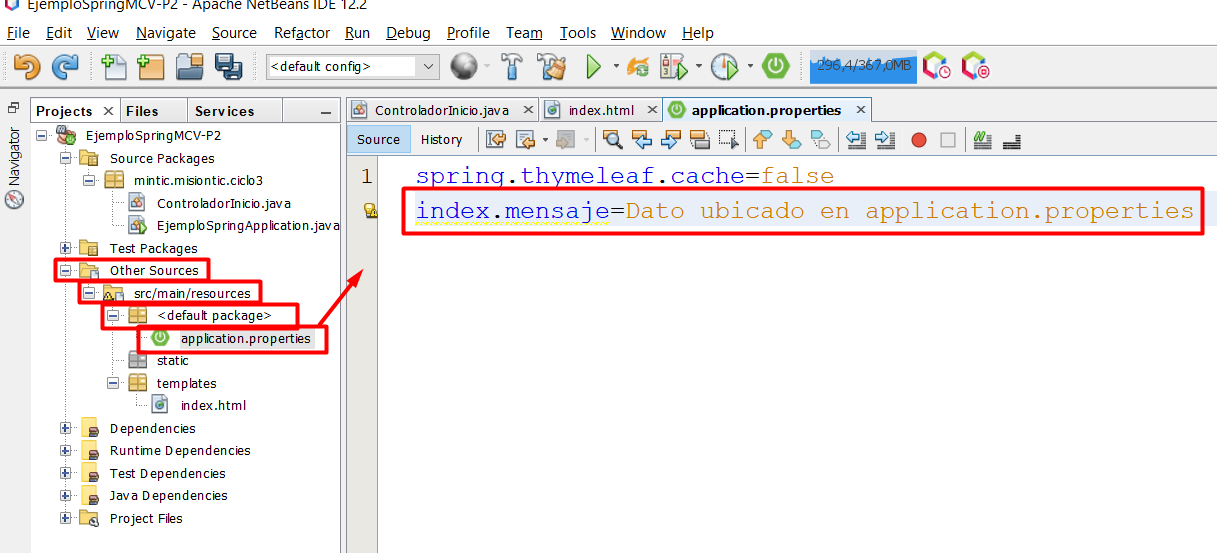
Actualizamos el navegador



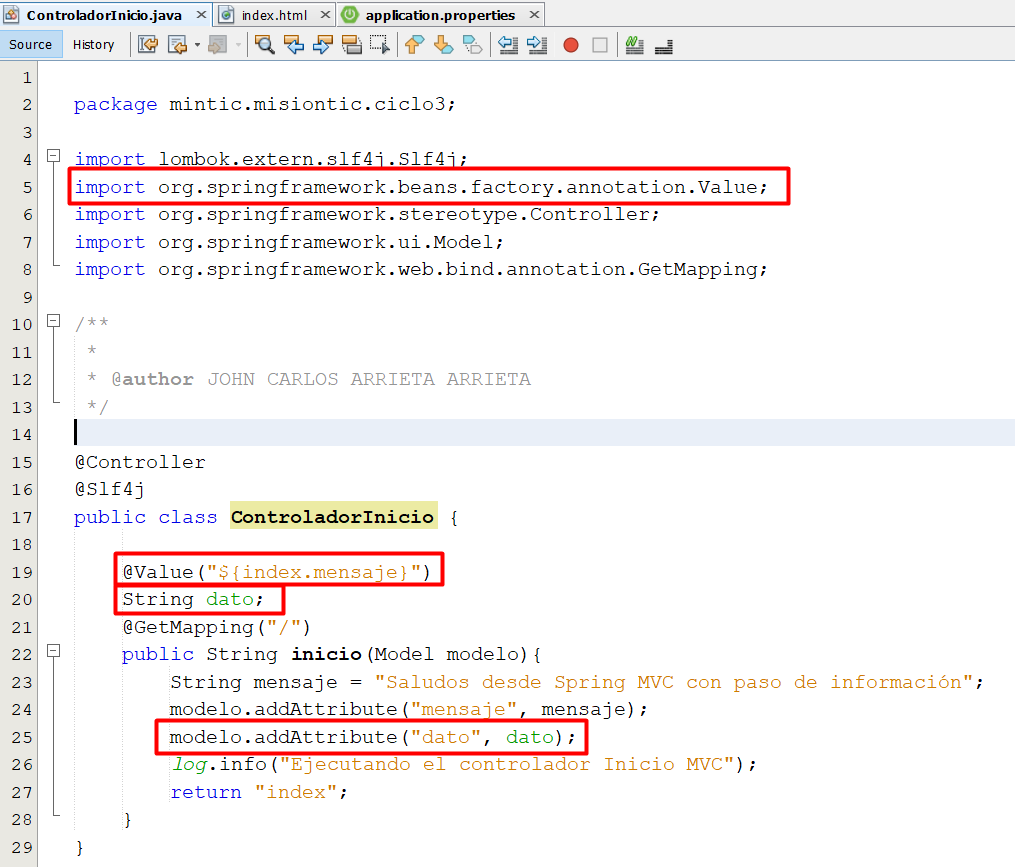
**Otra forma de pasar parámetros desde el controlador hacia la vista.**

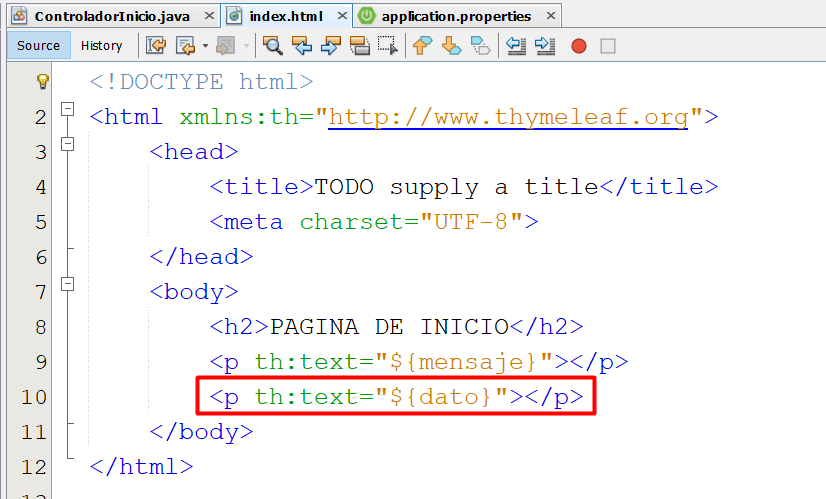
Esta nueva forma de pasar parámetros del controlador de vista utiliza las capacidades del archivo **Application.properties**

* Abrimos el archivo application.properties.
* Agregamos una nueva clave con su respectivo valor que deseamos compartir con la vista.
* Abrimos el archivo del controlador.
* Usamos la anotación **@Value** Incorporada en API de Spring.
* Esta anotación recibe como parámetros la clave que contiene el valor a compartir con la vista.
* Definimos una variable de tipo **String** sin valor por defecto, La anotación **@Value** Inyectará en la variable que acabamos de declarar el valor correspondiente a la clave que hemos agregado el archivo de propiedades.
* Modificamos el método **inicio** con el fin de agregarle un nuevo atributo a la variable modelo. Al igual que el caso anterior, este atributo puede contener cualquier clave.Y como valor la variable declarada anteriormente.
* Abrimos el archivo de la vista, el **index.html**.
* Le agregamos una nueva etiqueta en la que podamos recuperar el parámetro enviado usando la clave correspondiente.



Creamos la variable en aplication.properties para luego referenciarla con la notación @Value, con el fin de atrapar ese dato y añadirlo como atributo en nuestro motor de plantillas





Guardamos

ejecutamos

Actualizamos el navegador

