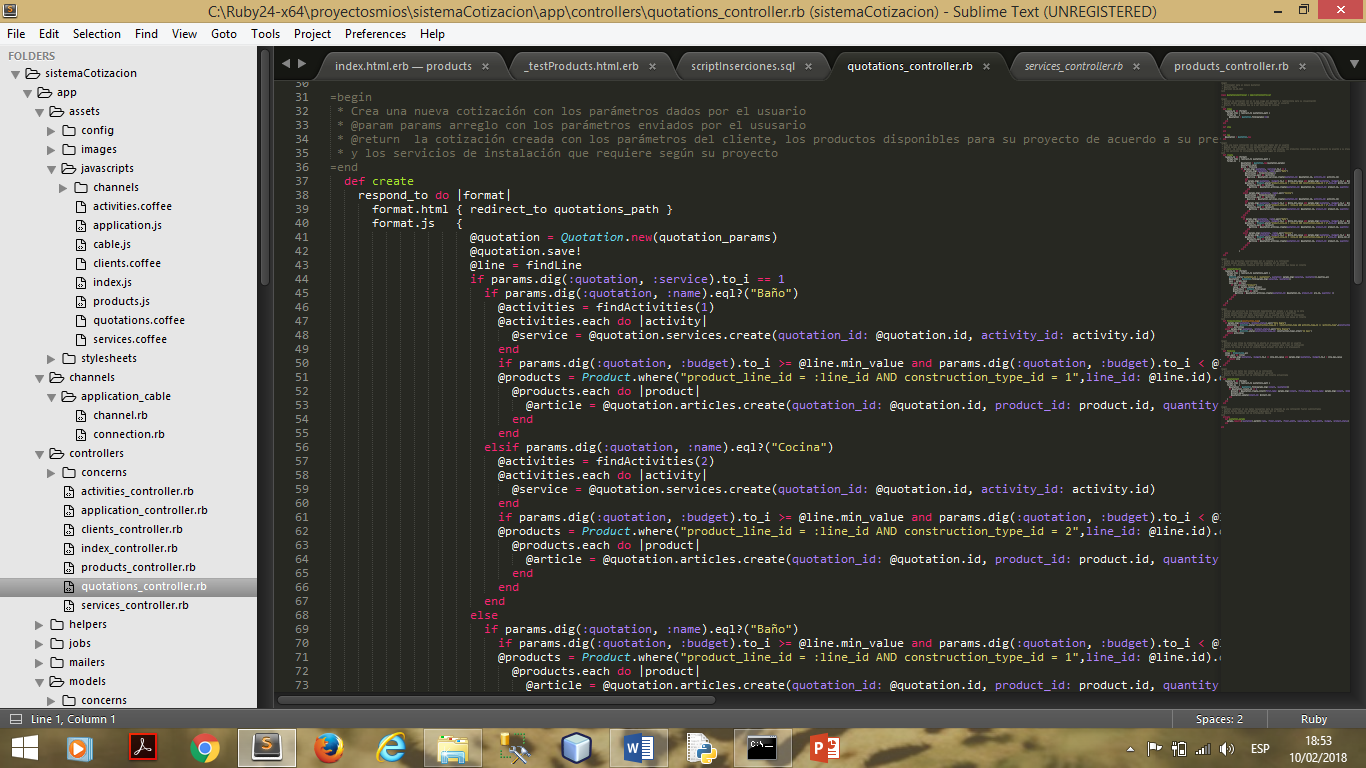
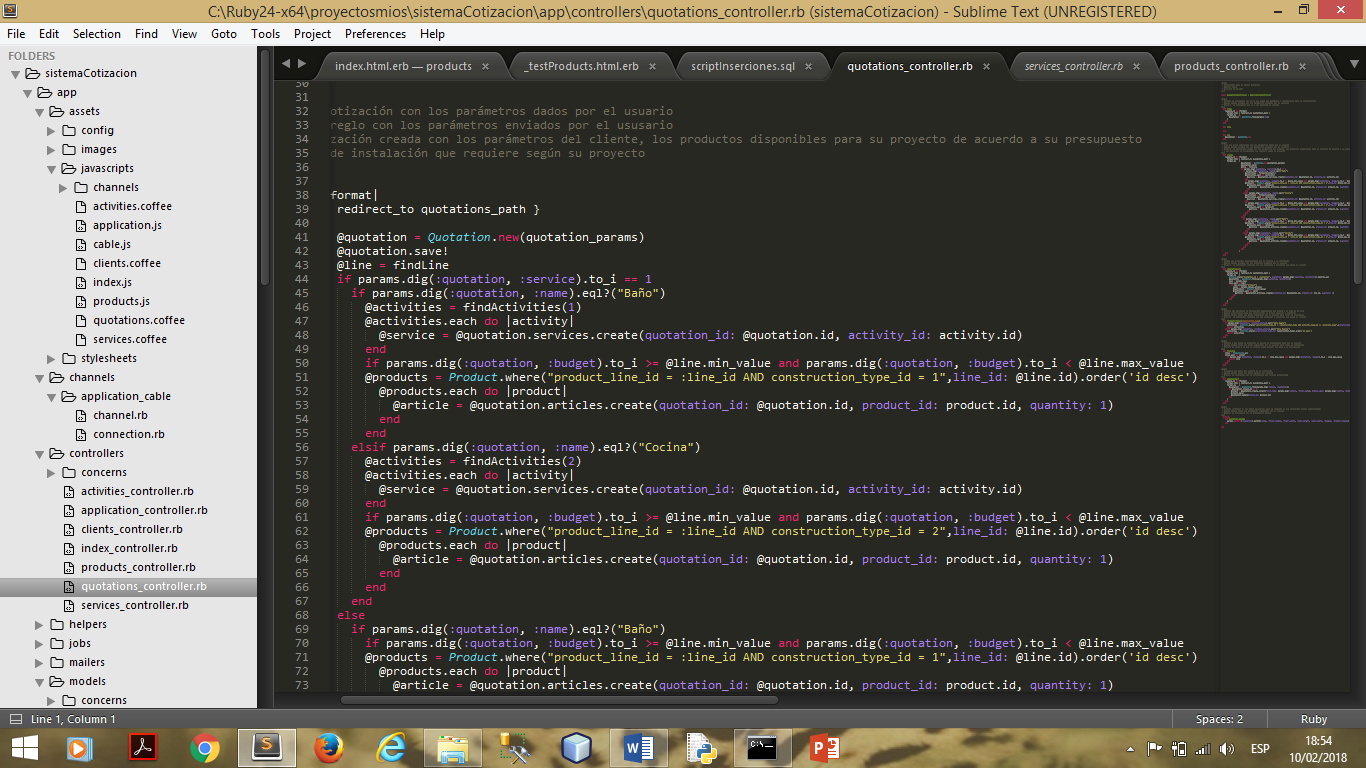
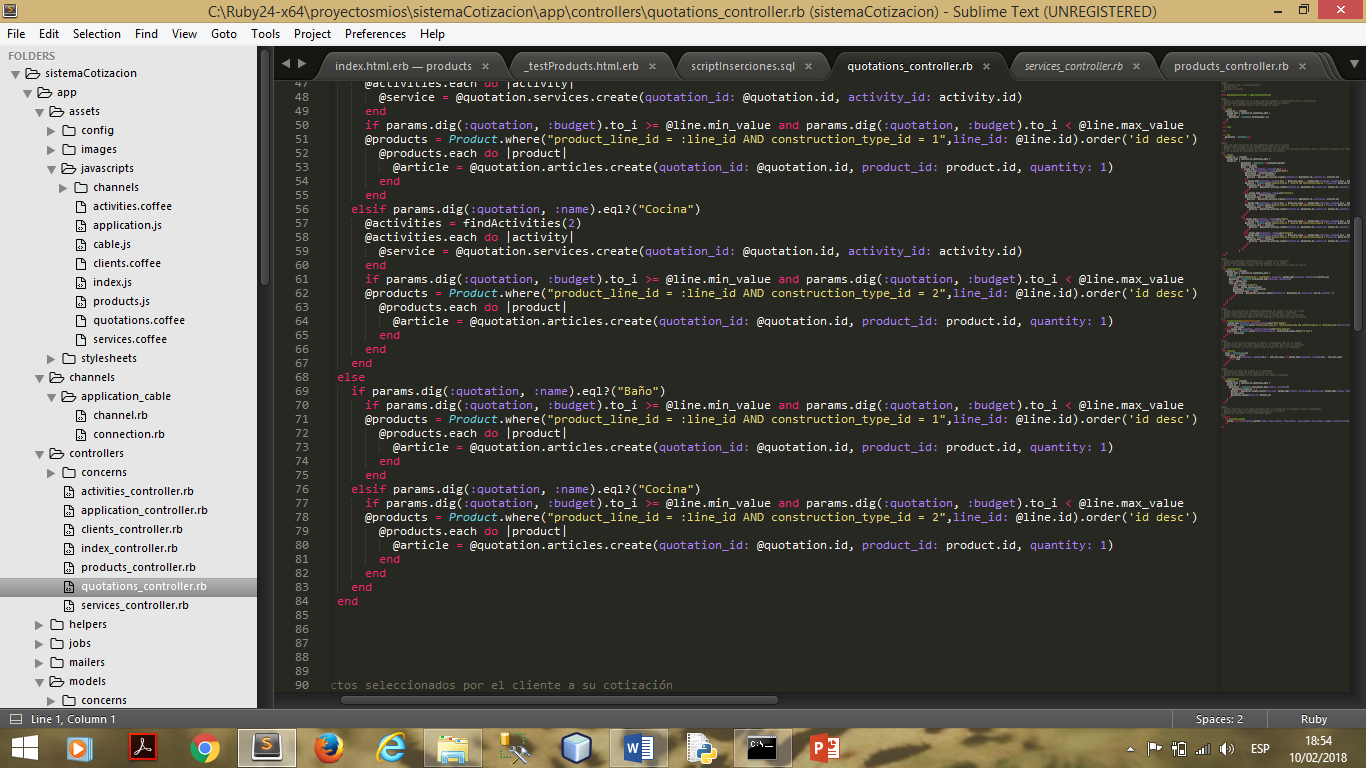
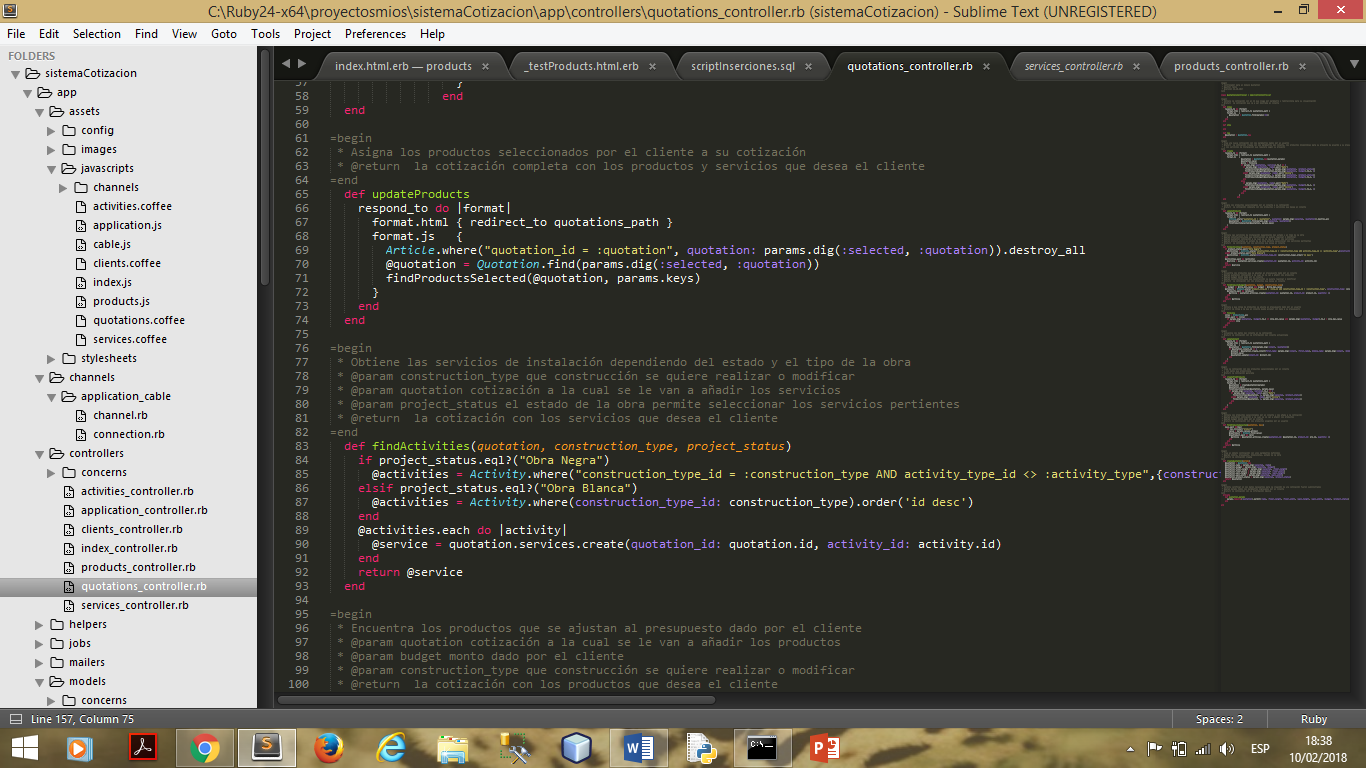
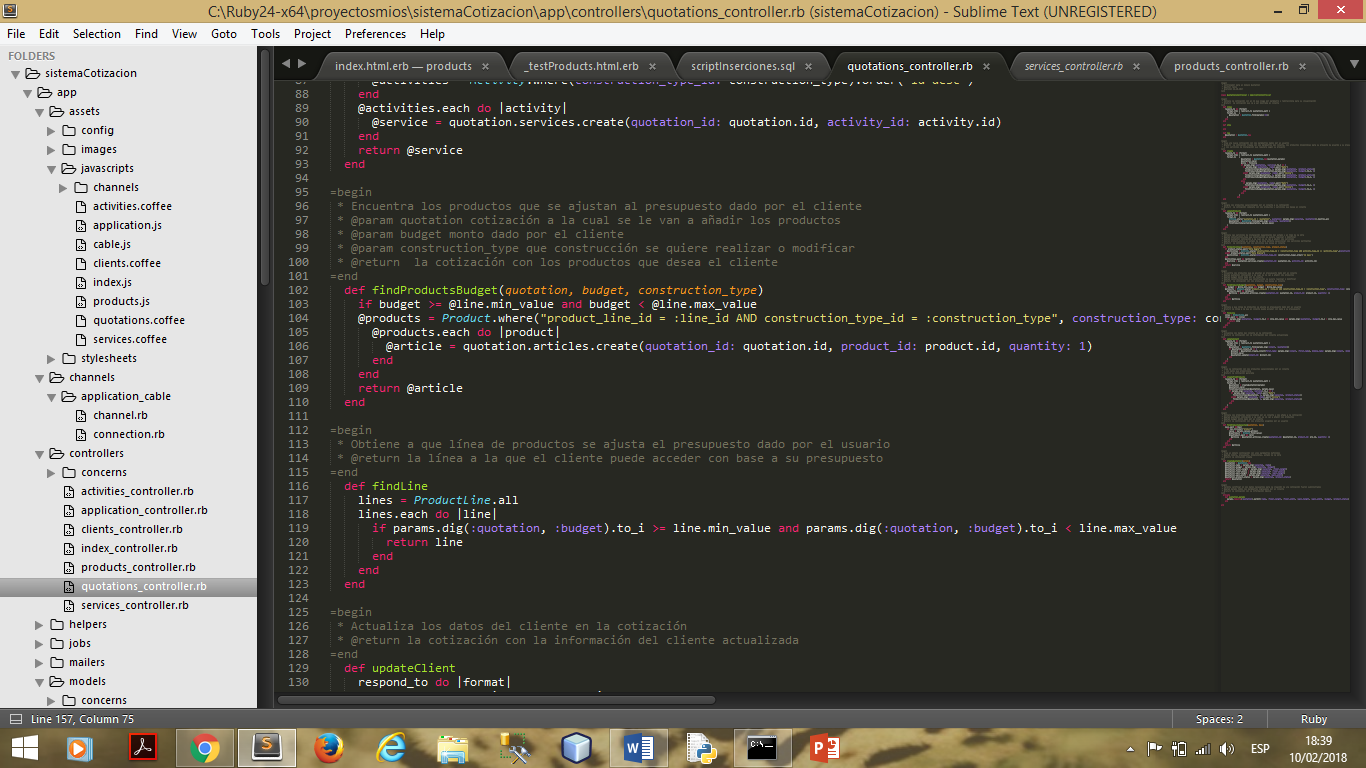
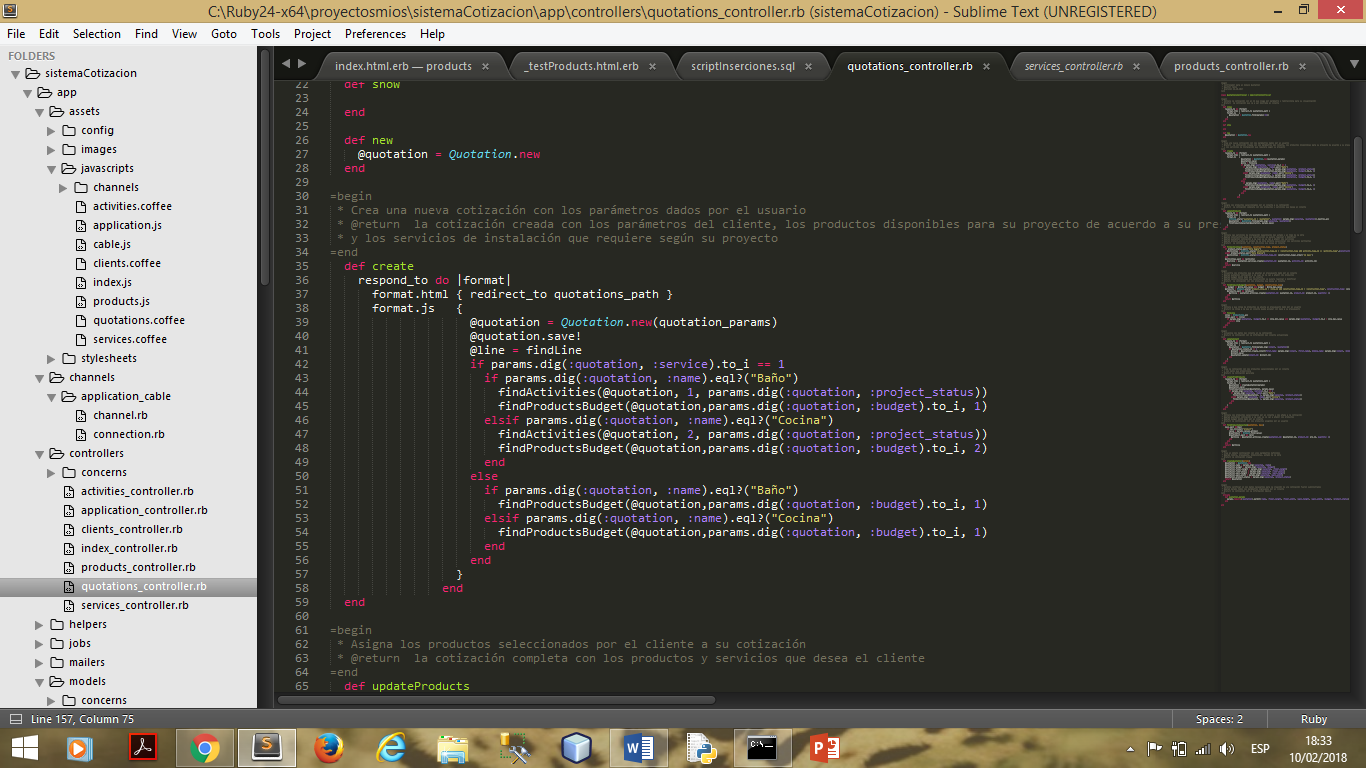
**COMPARATIVO REFACTORIZACIÓN**

Después de que se eligió refactorizar el método create, se procedió a realizar la refactorización y nos dimos cuenta de que era un método con muchas carga, este solo método era el que realizaba la cotización sin importar si era de baño o de cocina, tenía código duplicado ya que tenía acciones que debían de ejecutarse varias veces para poder realizar el cálculo y la elaboracion de las cotizaciones; a continuación se ve el código anterior del método create de la clase quotations\_controller:



Después de realizar la refactorización se crean dos métodos aparte del método create que son findActivities que es el encargado de obtener los servicios de instalación dependiendo del estado y el tipo de la obra para agregarlos a la cotización y tener las descripciones de cada una de las actividades que se requieren; a continuación vemos el método findActivities:

El otro método es el método findProductsBudget que es el encargado de encontrar los productos que se ajustan al presupuesto dado por el cliente, este método es muy importante para el proyecto planteado hacia corona ya que es básicamente la propuesta innovadora y diferenciadora de los demás proyectos del curso; elegir los productos a partir del presupuesto del cliente, veamos el método a continuación:

Por ultimo nos queda el método inicial que es el método create de la clase quotations\_controller, en este método lo se hace es crear una nueva cotización con los parámetros dados por el usuario, llamando a los dos métodos anteriores para poder realizar las búsquedas en la base datos de los datos importantes para la cotización que son los materiales, los servicios con sus respectivas actividades y los precios de cada uno de estos insumos; a continuación vemos el nuevo método create:

Como resultado encontramos que el código está mucho más limpio y más entendible aparte podemos notar que es mucho más corto que el código inicial y que estamos dando posibilidades futuras para el uso de los dos métodos creados, por lo tanto podemos concluir que fue una refactorización exitosa y necesaria, cabe destacar que la conclusión la sacamos después de las pruebas realizadas y de la comparación que se realiza en este documento.