**I. Class Main:**

* No posee atributos
* Metodos:

public static void main(String[] args)

Este método se ejecuta automaticamente al ejecutar el programa, llama al método ValidadorMenu(). No cuenta con entradas o salidas. Es de tipo public static

private static void Texto()

Este metodo tiene como salida por pantalla las opciones del menú de Main. No posee entradas. Es de tipo private static.

private static void ValidadorMenu()

Este metodo tiene como entrada a la opcion seleccionada posteriormente a que se llame a Texto(). El metodo cuenta con una validación de la opcion seleccionada. No posee salida. Es de tipo private static.

private static void OpcionElegida(int opc)

Metodo de tipo private, requiere el parametro opc de tipo int, este parametro proviene de ValidadorMenu(). OpcionElegida(int opc) es llamado dentro de ValidadorMenu(). OpcionElegida(int opc) cuenta con una validación de opc.

Se crea un objeto tipo Recetario, a fin de poder llamar a metodos de esa clase.

Este metodo se preocupa de evaluar el valor de opc y dependiendo de eso llamar al repectivo metodo de clase Recetario. VerCantidadRecetas() y ElegirReceta() , AgregarRecetas(), QuitarReceta(), Case5() y salir.

private static void Case5(Recetario recetario)

Tiene como parametro recetario. Es un menú que permite elegir entre buscar recetas ingresando un ingrediente o cambiar un determinado ingrediente. El metodo cuenta con sus validaciones. Después de validar, se llama a EvaluarOpcCase5(recetario, opcCase5).

private static void EvaluarOpcCase5(Recetario recetario, int opcCase5)

Tiene parametros que provienen de Case5(recetario). Este metodo evalua si se llama a MostrarAllRecetas() o a CambiarIngrediente(). Cuenta con un validador incorporado.

No devuelve datos.

**II. Class Receta:**

* Posee como atributos a ingredientes, un ArrayList<Ingrediente>, a instrucciones, un ArrayList<Instruccion> y a nombre, un String.
* Metodos:

Receta()

El constructor no necesita parametros, sirve para crear objetos de tipo Receta.

public void CrearIngredientes()

Este metodo pide al usuario si desea ingresar un ingrediente (IngresarIngrediente()) y posteriormente pide si desea, mas Ingredientes a una receta. Cuenta con las respectivas validaciones. No posee salida. Es public.

private void IngresarIngredientes()

Este metodo pide al usuario ingresar un Ingrediente a la Receta y lo agrega a ingredientes. No posee salida. Es private

public void MostrarIngredientes()

Este metodo recorre ingredientes y muestra los Ingredientes de la Receta. No posee entradas y es un metodo void de tipo public.

public int ObtenerViejoIngrediente()

Llama a MostrarIngredientes() de la Receta. Su función principal es obtener posicion de un ingrediente dentro del ArrayList ingredientes usando un Ingrediente ingresado por teclado y retornarla.

public void CrearInstruccion()

Este metodo no tiene parametros, llama a IngresarPaso() y pregunta si el usuario desea ingresar otro paso. Posee las respectivas validaciones. No tiene salidas. Es public.

private void IngresarPaso()

Este metodo no tiene parametros, pide al usuario un paso (Instruccion) de la Receta y lo añade a instrucciones. No posee salida. Es private.

public void MostrarInstrucciones()

Este metodo recorre instrucciones y muestra las Instrucciones de la Receta. No posee entradas y es un metodo void de tipo public.

*Los atributos cuentan con metodos get y set.*

**III. Class Recetario:**

* Posee como atributo a recetas, un ArrayList<Receta>.
* Metodos:

public void AgregarReceta()

Este metodo no tiene parametros, pide al usuario que ingrese una Receta, con su respectivo nombre, CrearIngredientes() y CrearInstruccion(), posteriormente se añade a recetas. No tiene salida. Es public.

public void QuitarReceta()

Este metodo no posee parametros, llama a Preguntar() y teniendo esa información eliminar al respectivo elemento en recetas. No tiene salidas. Es publico

private static int Preguntar()

Este metodo no posee parametros, llama a MostrarAllRecetas() y pide al usuario que ingrediente que quiere eliminar y retorna la posicion ingresada por el usuario. Es private static.

public static void MostrarAllRecetas()

Este metodo recorre recetas y muestra las Recetas. No posee entradas y es un metodo void de tipo public static.

public void FindRecetaIngredientes()

Este metodo pide al usuario que ingrese un Ingrediente y llama a Buscar() usando a la entrada como parametro de Buscar(). Es public. No posee salida.

private static void Buscar(String ing)

Este metodo pide al usuario que ingrese un determinado Ingrediente para buscarlo de recetas **(LUCAS QUE MAS LE PONGO AQUI)**

public void CambiarIng()

Metodo llamado desde EvaluarOpcCase5(recetario,opcCase5) en Main. Busca una receta ingresada por el usuario en recetas, realiza una validación y finalmente pide al usuario el nombre del nuevo Ingrediente y reemplaza el valor anterior. Es public void.

public void VerCantidadRecetas()

Este metodo muestra al usuario la cantidad de recetas disponibles en recetas. Es public y no posee entradas ni salidas.

public void ElegirReceta()

No requiere entradas ni posee salidas. Este metodo llama a MostrarAllRecetas() y luego pide al usuario que eliga una receta y llama a MostrarOpcionElegida(recetaElegida) con el valor ingresado como parametro del metodo.

private static void MostrarOpcionElegida(int recetaElegida)

Requiere como parametro recetaElegida, dato obtenido de ElegirReceta(). Este metodo muestra por pantalla el nombre de una receta, sus ingredientes (MostrarIngredientes()) y sus instrucciones (MostrarInstrucciones()). Es private static void.

**IV. Class Instruccion:**

* Posee como atributo a paso, un String.
* Metodos:

public Instruccion()

Este metodo se crea objetos de tipo Instrucción e inicia a paso.

*Posee metodos get y set del atributo paso.*

**V. Class Ingrediente:**

* Posee como atributo a nombre, un String.
* Metodos:

public Ingrediente()

Este metodo se crea objetos de tipo Ingrediente e inicia a nombre.

*Posee metodos get y set del atributo nombre.*