**I. Class Main:**

* No posee atributos
* Métodos:

public static void main(String[] args)

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

private static void Texto()

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

private static void ValidadorMenu()

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

private static void OpcionElegida(int opc)

Método de tipo private, requiere el parámetro opc de tipo int, este parámetro 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 métodos de esa clase.

Este método se preocupa de evaluar el valor de opc y dependiendo de eso llamar al repectivo método de clase Recetario. VerCantidadRecetas() y ElegirReceta() , AgregarRecetas(), QuitarReceta(), Case4() y salir.

private static void Case4(Recetario recetario)

Tiene como parámetro recetario. Es un menú que permite elegir entre buscar recetas ingresando un ingrediente o cambiar un determinado ingrediente. El método cuenta con sus validaciones. Después de validar, se llama a EvaluarOpcCase4(recetario, opcCase4).

private static void EvaluarOpcCase4(Recetario recetario, int opcCase4)

Tiene parámetros que provienen de Case4(recetario). Este método 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.
* Métodos:

Receta()

Es el método constructor no necesita parámetros, sirve para crear objetos de tipo Receta.

public void CrearIngredientes()

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

private void IngresarIngredientes()

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

public void MostrarIngredientes()

Este método recorre ingredientes y muestra los Ingredientes de la Receta. No posee entradas y es un método 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 método no tiene parámetros, 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 método no tiene parámetros, pide al usuario un paso (Instruccion) de la Receta y lo añade a instrucciones. No posee salida. Es private.

public void MostrarInstrucciones()

Este método recorre el ArrayList instrucciones y muestra las instrucciones de la Receta. No posee entradas y es un método void de tipo public.

*Los atributos cuentan con métodos get y set.*

**III. Class Recetario:**

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

public void AgregarReceta()

Este método no tiene parámetros, 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 método no posee parámetros, llama a Preguntar() y teniendo esa información elimina al respectivo elemento en recetas. No tiene salidas. Es public.

private static int Preguntar()

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

public static void MostrarAllRecetas()

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

public void FindRecetaIngredientes()

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

private static void Buscar(String ing)

Este método es llamado por FindRecetaIngrediente(). Requiere de un parámetro String que es el nombre del ingrediente para poder comparar y mostrar en pantalla los elementos del ArrayList recetas que poseen dicho String. Es private.

public void CambiarIng()

Método llamado desde EvaluarOpcCase4(recetario,opcCase4) 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 método 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 método llama a MostrarAllRecetas() y luego pide al usuario que elija una receta y llama a MostrarOpcionElegida(recetaElegida) con el valor ingresado como parámetro del método.

private static void MostrarOpcionElegida(int recetaElegida)

Requiere como parémetro recetaElegida, dato obtenido de ElegirReceta(). Este método 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.
* Métodos:

public Instruccion()

Este método crea objetos de tipo Instrucción e inicia a paso.

*Posee métodos get y set del atributo paso.*

**V. Class Ingrediente:**

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

public Ingrediente()

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

*Posee métodos get y set del atributo nombre.*