El ejecutivo

Lenguajes de programación – Andrés Felipe Penagos

Contenido

[1. Integrantes del equipo: 1](#_Toc33878011)

[2. Nombre de la empresa: 1](#_Toc33878012)

[3. Logo de la empresa: 1](#_Toc33878013)

[4. Descripción del problema: 1](#_Toc33878014)

[5. Propuesta de valor: 1](#_Toc33878015)

[6. Stakeholders internos: 1](#_Toc33878016)

[7. Stakeholders externos: 1](#_Toc33878017)

[8. Listado requerimientos funcionales: 1](#_Toc33878018)

[9. Requerimientos no funcionales: 4](#_Toc33878019)

[10. Diseño: 4](#_Toc33878020)

[a) Mockup: 4](#_Toc33878021)

## Integrantes del equipo:

Andrés Felipe Penagos Rodriguez

## Nombre de la empresa:

EL EJECUTIVO

## Logo de la empresa:



## Descripción del problema:

Según datos de la firma Radar los hogares compraron en almuerzos corrientes $1,47 billones. ¿Pero cómo operan estos restaurantes? Por lo general estos establecimientos se encuentran en pequeños locales, y muchos surgen como un rebusque o un desvare. Los pedidos son tomados por meseros en libretas, lo que ocurre que muchas veces no se le haga seguimiento al pedido y llegue tarde, confundan el pedido o el dueño no tenga información de lo que se está vendiendo diariamente.

## Propuesta de valor:

El ejecutivo busca resolver un vacío de información en estos establecimientos, que les permita de forma práctica solicitar, tomar, cobrar y entregar pedidos por medio de un aplicativo digital, que les permita visualizar en tiempo real los estados de los pedidos, con base a esta información les permita tomar decisiones enfocadas aumentar el nivel de servicio y así mismo sus ingresos.

## Stakeholders internos:

1. Meseros: son aquellos que solicitan los pedidos de los comensales.
2. Cocineros: Son aquellos de cocinar los pedidos, de ellos dependen la duración de entrega de un pedido.
3. Cajeros: los cajeros son los encargados de cobrar los pedidos solicitados.
4. Administrador: es el perfil habilitado para realizar las anteriores funciones y modificar los perfiles anteriores.

## Stakeholders externos:

1. Proveedores: son aquellos que proveen al negocio de los insumos para la cocción de los productos.
2. Clientes: son los que se verán beneficiados al tener una mejor gestión de sus pedidos.
3. Arrendador: el arrendador podrá ver información de ventas que le genere confianza en el arrendatario

## Listado requerimientos funcionales:

|  |  |
| --- | --- |
| IDENTIFICADOR: R1 | NOMBRE: Creación de Usuarios |
| TIPO: (NECESARIO/DESEABLE) NECESARIO | PRIORIDAD DE DESARROLLO: ALTA |
| ENTRADA: \* NOMBRE DE EMPLEADO \*CARGO DE EMPLEADO \*CODIGO EMPLEADO \*CONTRASEÑA | SALIDA: CRUD Usuario. |
| DESCRIPCION: La aplicación permitirá crear, consultar, actualizar y eliminar nuevos usuarios. Los usuarios son únicamente los empleados del restaurante. | |

|  |  |
| --- | --- |
| IDENTIFICADOR: R2 | NOMBRE: Sub Producto |
| TIPO: (NECESARIO/DESEABLE) NECESARIO | PRIORIDAD DE DESARROLLO: ALTA |
| ENTRADA: \*Nombre subproducto  \*código subproducto | SALIDA: creación, lectura, actualización y eliminación del subproducto. |
| DESCRIPCION: Los subproductos son aquellos componentes que integran un menú. Por ejemplo, un menú lo compone la sopa, el principio y el jugo, en este caso la sopa seria un subproducto. | |

|  |  |
| --- | --- |
| IDENTIFICADOR: R3 | NOMBRE: Menú |
| TIPO: (NECESARIO/DESEABLE) NECESARIO | PRIORIDAD DE DESARROLLO: ALTA |
| ENTRADA: \* CODIGO MENU \*NOMBRE MENU \*CODIGOS SUBPRODUCTOS \*NOMBRES SUBPRODUCTOS \*PRECIO | SALIDA: creación, lectura, actualización y eliminación del menú. |
| DESCRIPCION: El menú es considerados como una combinación de sub productos como por ejemplo, el menú esta compuesto de sopa, carne asada, arroz, frijol, ensalada y jugo de mora. El mesero podrá realizar diferentes combinaciones y así crear un nuevo menú, o modificar -eliminar algún subproducto. | |

|  |  |
| --- | --- |
| IDENTIFICADOR: R4 | NOMBRE: Pedido |
| TIPO: (NECESARIO/DESEABLE) NECESARIO | PRIORIDAD DE DESARROLLO: ALTA |
| ENTRADA: \* NOMBRE MENU \*CANTIDAD DE MENUS \*TOTAL PRECIO POR MENUS \*ESTADO DEL PEDIDO \*TOTAL PRECIO PEDIDO \*FECHA NOVEDAD ESTATUS PEDIDO | SALIDA: creación, lectura, actualización y eliminación del pedido. |
| DESCRIPCION: Los pedidos son el número de menús que desea el cliente, el cual solicita y entrega el mesero, cocina el cocinero y cobra el cajero. Un pedido puede contener 2 menús 1 y 1 menú 2. | |

|  |  |
| --- | --- |
| IDENTIFICADOR: R5 | NOMBRE: Asociación de perfiles |
| TIPO: (NECESARIO/DESEABLE) NECESARIO | PRIORIDAD DE DESARROLLO: ALTA |
| ENTRADA: \*Descripción de permisos por perfil. | SALIDA: Visualización de ventanas y CRUD por perfiles |
| DESCRIPCION: La visualización de las ventanas y el registro, actualización, lectura y eliminación de cada uno de los elementos de las ventanas se realizará de acuerdo al perfil de permisos del usuario. Por ejemplo, el mesero solo podrá solicitar y entregar pedidos, el cocinero solo podrá tomar y cocinar pedidos, y el cajero solo cobrar pedidos, y podrán visualizar las respectivas ventanas según su perfil | |

## Requerimientos no funcionales:

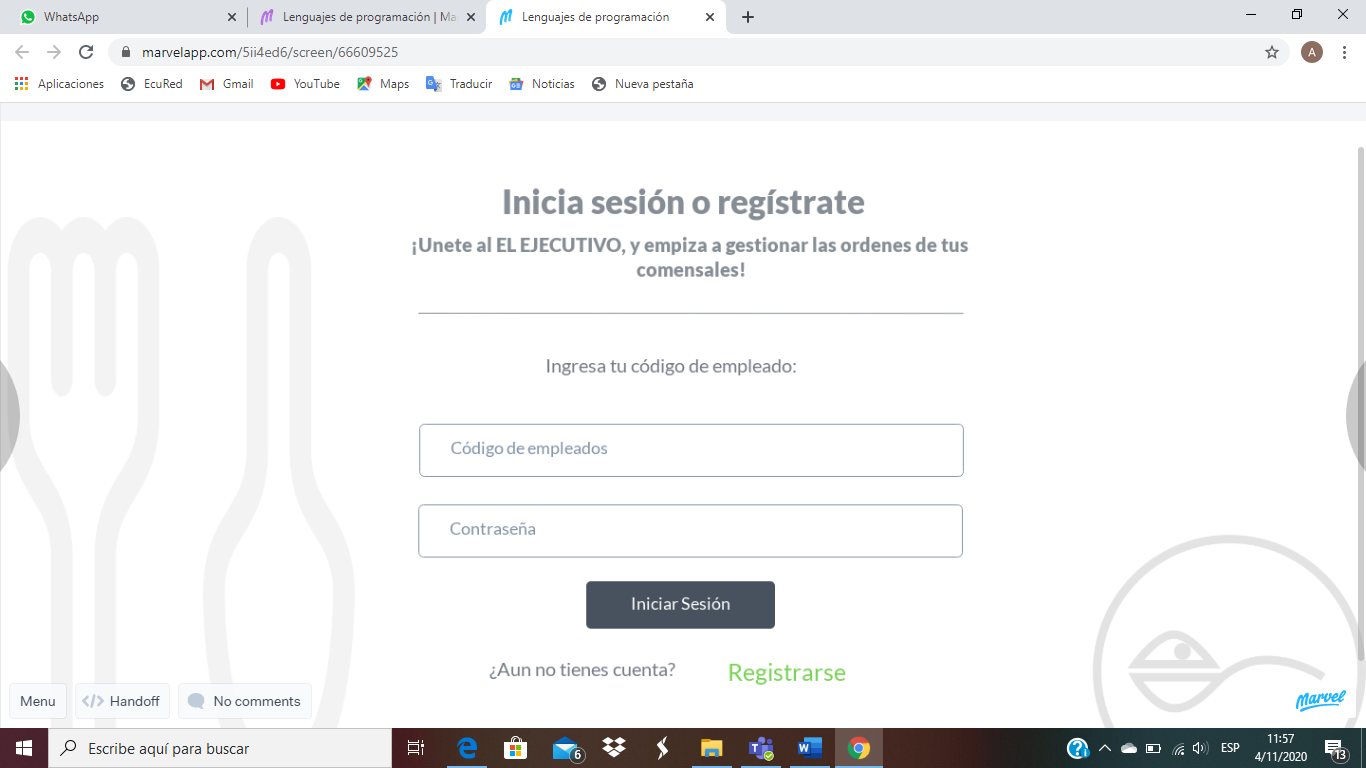
|  |  |
| --- | --- |
| IDENTIFICADOR: RN1 | NOMBRE: Interfaz Grafica |
| TIPO: (NECESARIO/DESEABLE) NECESARIO | PRIORIDAD DE DESARROLLO: ALTA |
| ENTRADA: \*concepto de la interfaz | SALIDA: Visualización interfaz grafica |
| DESCRIPCION: La interfaz grafica será realizada con la librería swing. | |

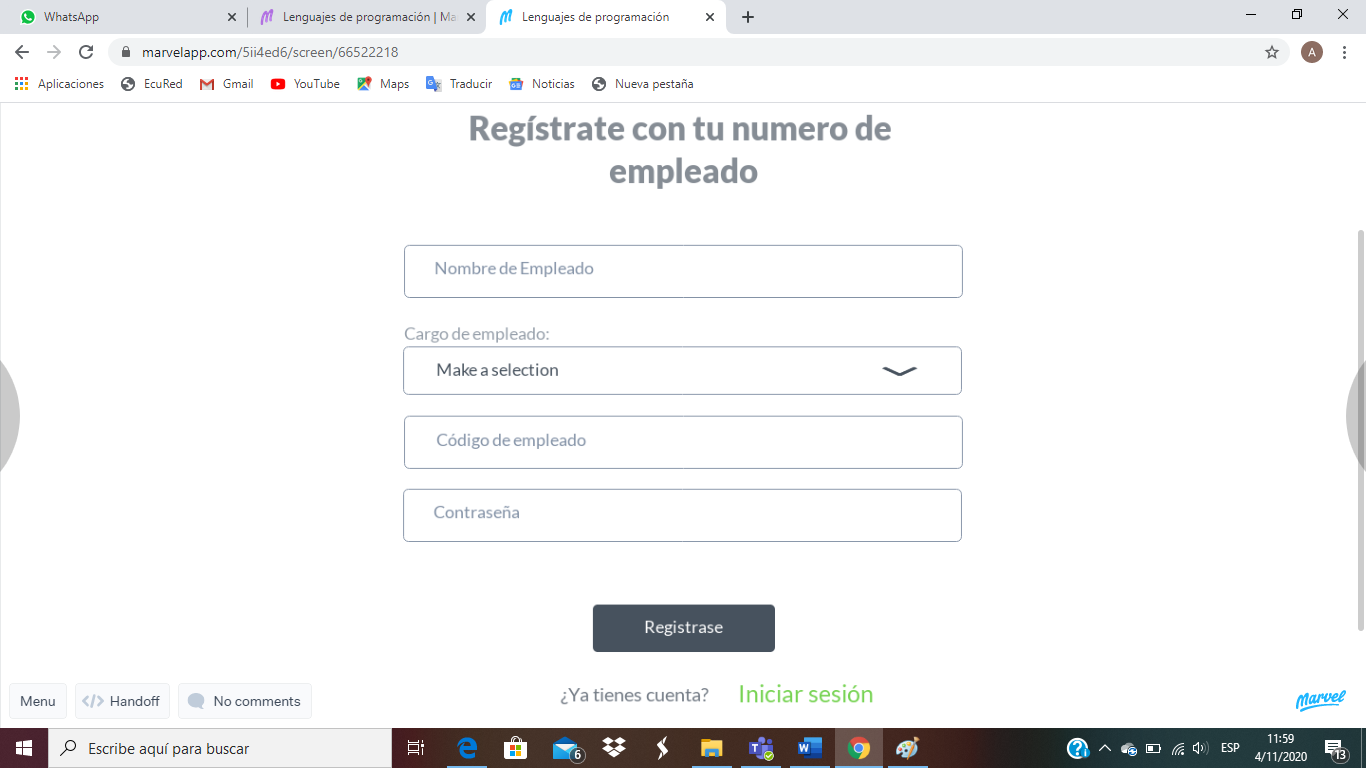
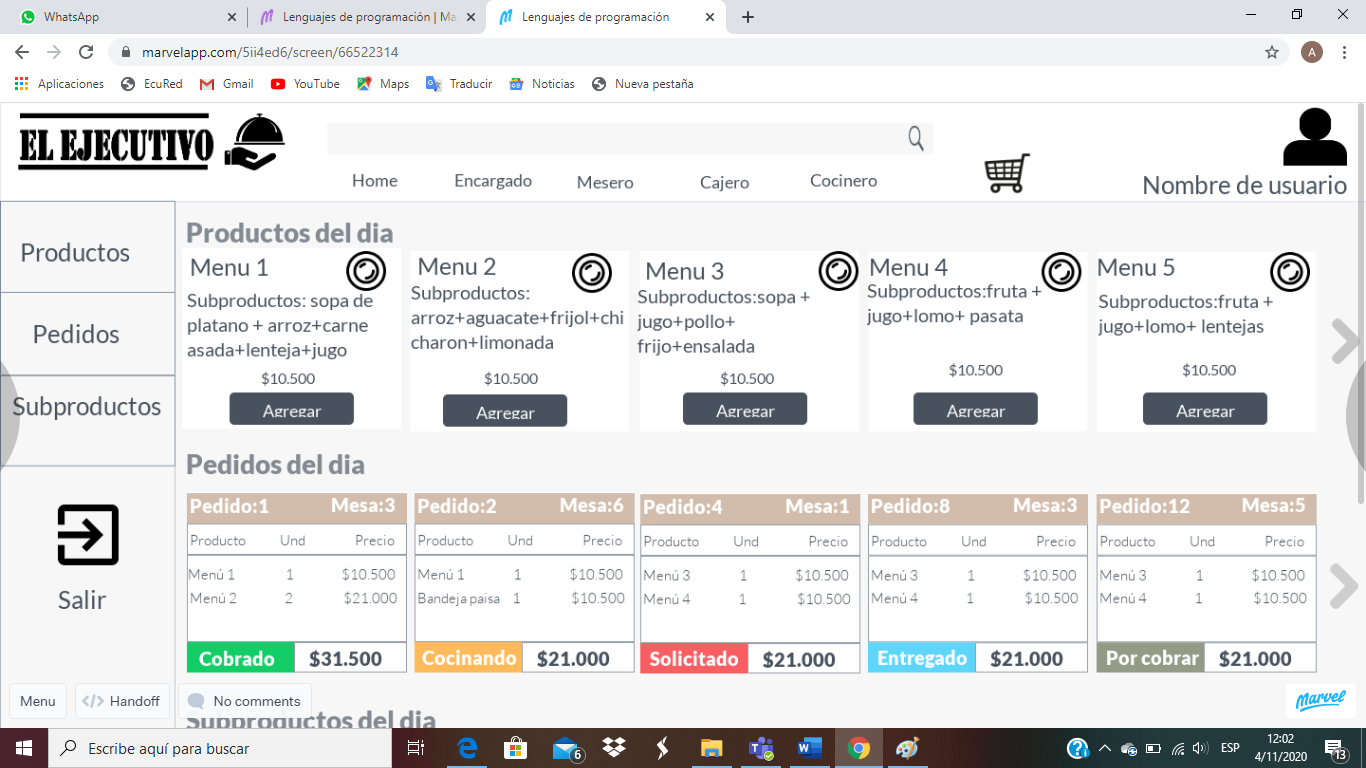
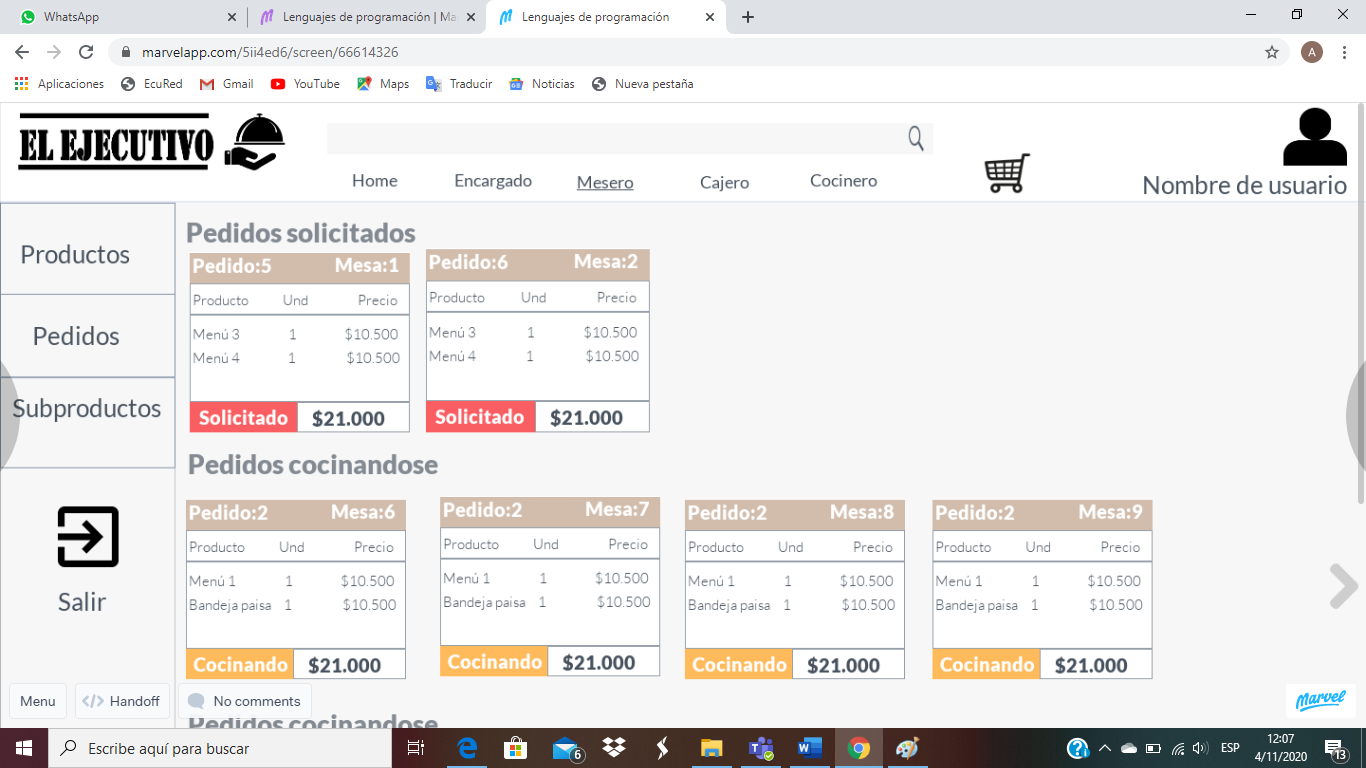
|  |  |
| --- | --- |
| IDENTIFICADOR: RN2 | NOMBRE: Protección de datos |
| TIPO: (NECESARIO/DESEABLE) NECESARIO | PRIORIDAD DE DESARROLLO: ALTA |
| ENTRADA: \*Diagnostico de los datos sensibles a hackeos | SALIDA: Diseño de seguridad de protección de datos |
| DESCRIPCION: Modelo de seguridad de datos. | |

|  |  |
| --- | --- |
| IDENTIFICADOR: RN3 | NOMBRE: usabilidad y eficiencia |
| TIPO: (NECESARIO/DESEABLE) NECESARIO | PRIORIDAD DE DESARROLLO: ALTA |
| ENTRADA: \*Diagnostico de funcionamiento de la operación del restaurante. | SALIDA: Modelamiento de la aplicación de uso sencillo y eficiente |
| DESCRIPCION: La aplicación será diseñada de acuerdo a las necesidades del restaurante y su operación, acompañada de un código ligero que le permita ser eficiente. | |

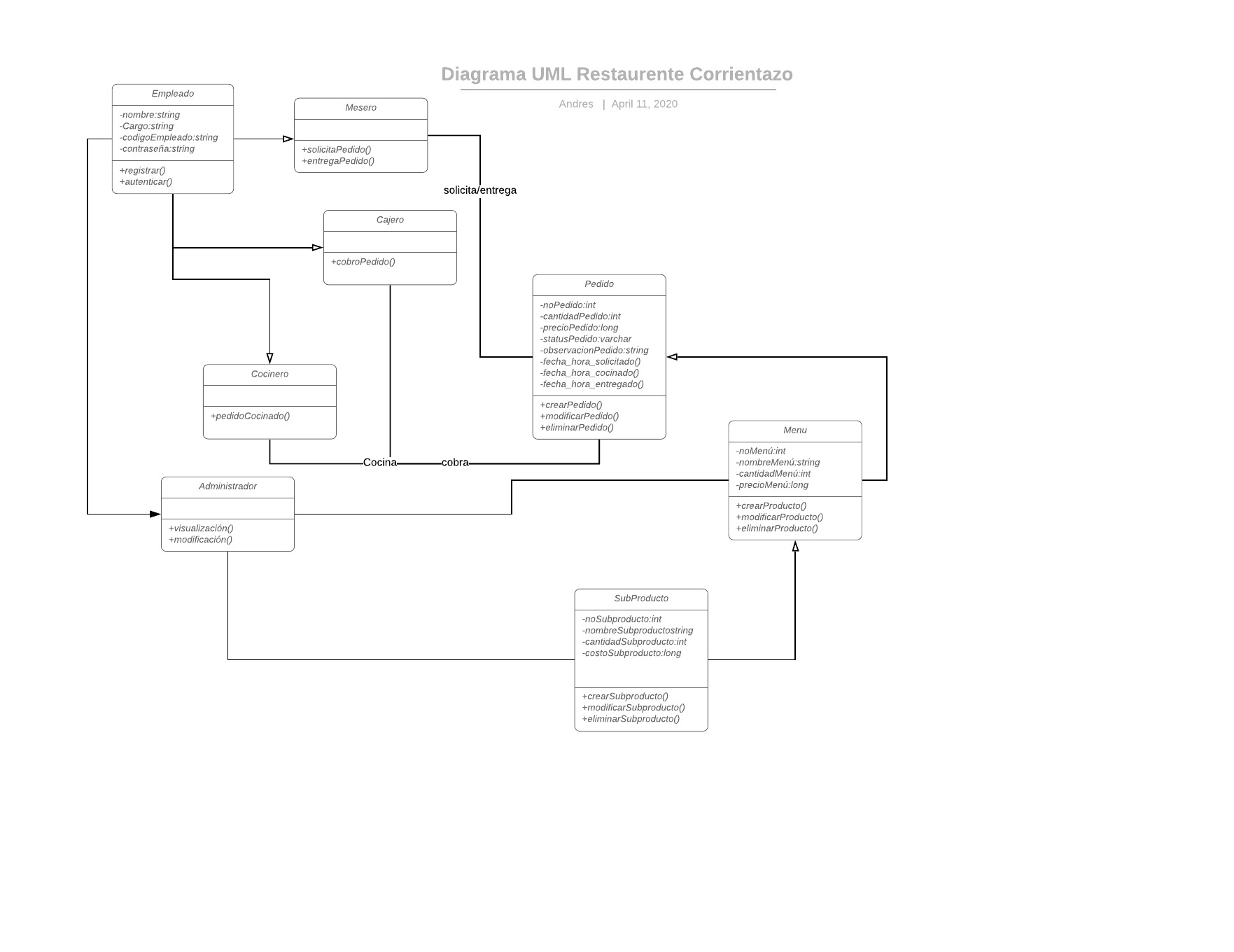
## Diseño Mockup:

1. Pagina de inicio



1. Pagina de Registro
2. HOME
3. MESERO

## DIAGRAM UML



public class Principal {  
      
    public static int  sumar (int a, int b) {  
        return a+b;      
    }  
      
    public static int  resta (int a, int b) {  
        return a-b;      
    }  
      
    public static int  multiplicacion (int a, int b) {  
        return a\*b;      
    }  
      
    public static float  division (float a, float b) {  
        float div=0;  
        try {  
            div= a/b;  
        } catch (ArithmeticException e) {  
            System.out.print("no se puede dividir entre cero");  
        }  
        return div;      
    }  
      
    public static void main(String[] args) {  
        System.out.println("Suma= "+ sumar(10, 20));  
        System.out.println("Resta= "+resta(10, 20));  
        System.out.println("Multiplicacion= "+ multiplicacion(10, 20));  
        System.out.println("Division= " + division(10, 5));  
        

    }

}