
PROYECTO PROGRAMACIÓN DIADAM



Tabla de contenido

Requisitos funcionales	4
Requisitos no funcionales.....	4
Requisitos de Información.....	5
Diagramas de casos de uso.....	5
Casos de uso.....	6
Diagrama de clases.....	15
Diagrama de entidad/relación	16
Patrones.....	16
Tareas ejercidas por cada componente del equipo	18
COSTE DEL PROYECTO	21

TITULO

DiaDam

INTEGRANTES DE EQUIPO

Jorge, Álvaro, Iván.

INTRODUCCIÓN

DiaDam, es un proyecto educativo de software desarrollado por alumnos del Ciclo Formativo de Grado Superior “Desarrollo de Aplicaciones Multiplataforma”.

Se pide simular cómo sería el desarrollo de una aplicación en un entorno laboral utilizando los conceptos enseñados en el módulo de programación.

Los requisitos técnicos del proyecto y los contenidos están fijados por José Luis González Sánchez, profesor del módulo de programación, de entornos y tutor del grupo.

DESCRIPCIÓN DE LA APP

Diseñaremos una aplicación que permita que el cliente pueda encargar productos a través de un menú seleccionándolos y podrá pagar desde la aplicación.

También la aplicación tendrá una parte para administrar los productos en la que solo se podrá meter el administrador, en esta parte se consultará los datos de los productos y se podrá realizar cambios en el menú del cliente.

ANÁLISIS

Requisitos funcionales

RF1-El cliente para poder comprar productos debe estar registrado y haber iniciado sesión antes.

RF2-El cliente deberá tener un carrito con todos los productos que desea comprar y también podrá quitar productos de ese mismo carro.

RF3-El cliente podrá ver una vista general de productos.

RF4-El cliente podrá salir de la aplicación.

RF5-El administrador podrá actualizar los productos.

RF6-El administrador podrá insertar productos.

RF7-El administrador podrá eliminar los productos.

RF9-El cliente podrá buscar podrá ordenar por id, nombre, precio, cantidad.

RF10-El cliente y el administrador podrán cambiar de estilo de la página.

RF11-El cliente podrá imprimir factura.

RF12- El administrador realiza un backup de las ventas.

RF13- El administrador podrá restaurar.

Requisitos no funcionales

RFN01-El administrador no podrá poner un precio menor o igual a 0 euros.

RFN02-El cliente tiene que dejar un email valido.

RFN03-No se podrá comprar más productos de los que hay en el Stock.

RFN04-La copia de seguridad es un json.

RFN05-La base de datos es en sqlite.

RFN06-El cliente no podrá comprar sin haber iniciado sesión.

RFN07-El cliente al registrarse debe introducir todos los datos.

RFN08- Imprimir factura html.

RFN09- Interfaz de usuario:

RFN09.1- Varios temas

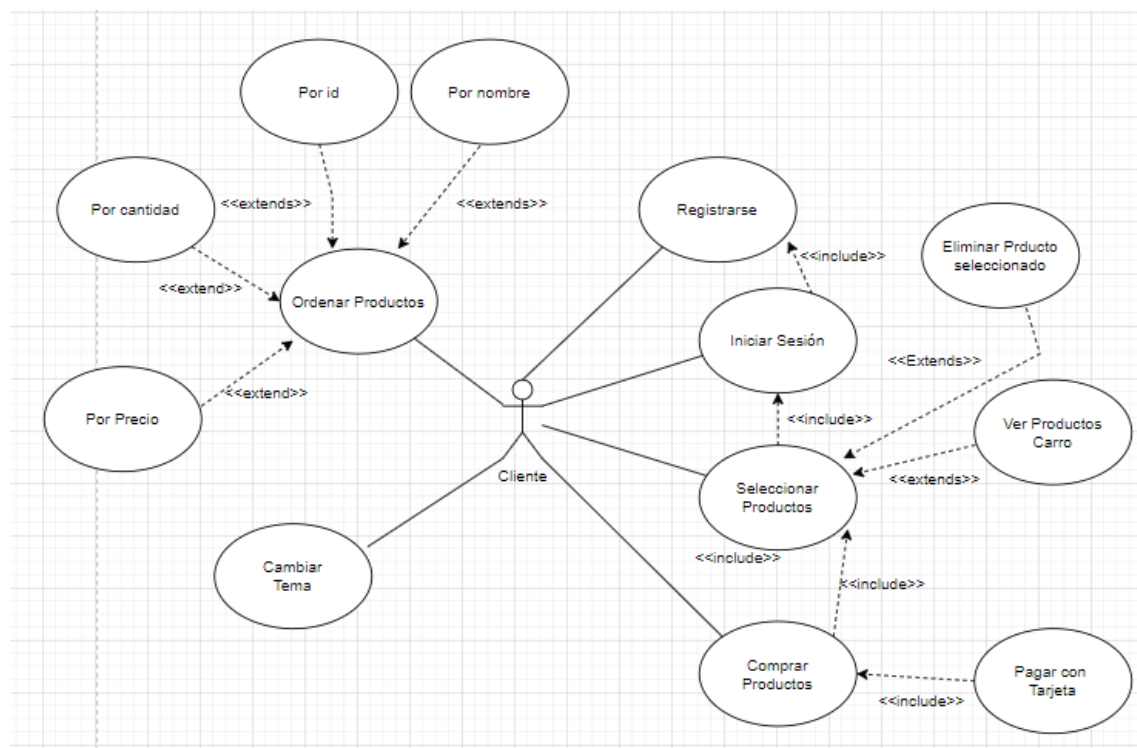
RFN09.2- Opción de mostrar carrito

Requisitos de Información

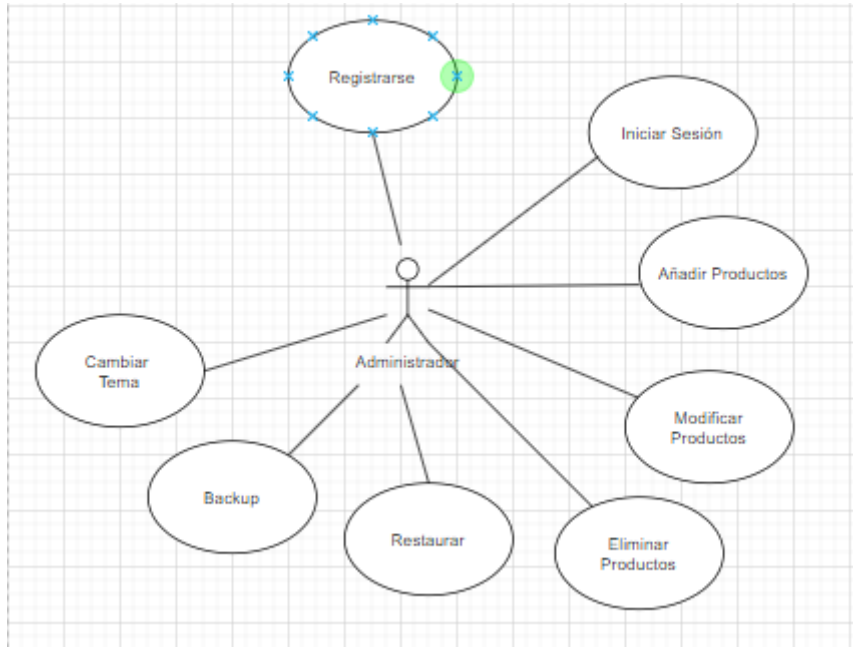
Manejaremos la información de los productos que estarán en la app y los usuarios que se hayan registrado.

Diagramas de casos de uso

Actor: Cliente.



Actor: Administrador.



Casos de uso

NOMBRE: Registro

ID:CU1

DESCRIPCIÓN: El cliente o el administrador se registra.

ACTORES: Cliente

PRECONDICIONES: Se requiere que el cliente no este registrado.

CURSO NORMAL:

1. El cliente o el administrador solicita registrarse.
2. El sistema o el administrador solicita los datos del cliente.
3. El cliente o el administrador rellena el formulario.
4. El sistema valida los datos introducidos.
5. El sistema muestra un mensaje de confirmación por pantalla.
6. El sistema registra al cliente en la base de datos.

POSTCONDICIONES:

El sistema registra al cliente en la base de datos.

NOMBRE: Iniciar Sesión

ID:CU1

DESCRIPCIÓN: El administrador o cliente solicita inicio de sesión.

ACTORES: Administrador, Cliente

PRECONDICIONES: Se requiere que el administrador o cliente este registrado en el sistema.

CURSO NORMAL:

1. El administrador o cliente solicita iniciar sesión.
2. El sistema solicita los datos del administrador o cliente.
3. El administrador o cliente rellena el formulario con los datos de correo y contraseña.
4. El sistema valida los datos introducidos.
5. El sistema comprueba si es cliente o administrador.

6. El sistema muestra un mensaje de confirmación por pantalla.

6.1. Si es administrador abrir ventana de administrador.

6.2. Si es cliente abrir ventana de cliente.

POSTCONDICIONES:

El cliente o el administrador realiza la acción deseada.

NOMBRE: Cerrar sesión

ID:CU2

DESCRIPCIÓN: El cliente o Administrador cierra sesión.

ACTORES: Cliente, Administrador

PRECONDICIONES: Se requiere que el cliente o el administrador haya iniciado sesión.

CURSO NORMAL:

1. El cliente o administrador solicita cerrar sesión.

2. El sistema cierra la sesión.

POSTCONDICIONES:

Se cerró la aplicación.

NOMBRE: Cambiar tema

ID:CU3

DESCRIPCIÓN: El administrador o el cliente solicita inicio de sesión.

ACTORES: Administrador, Cliente

PRECONDICIONES: Se requiere que el administrador o cliente este registrado en el sistema.

CURSO NORMAL:

1. El administrador o cliente solicita iniciar sesión.

2. El sistema solicita los datos del administrador o cliente.

3. El administrador o cliente rellena el formulario con los datos de correo y contraseña.

4. El sistema valida los datos introducidos.
5. El sistema comprueba si es cliente o administrador.
6. El sistema muestra un mensaje de confirmación por pantalla.
- 6.1. Si es administrador abrir ventana de administrador 8.
- 6.2. Si es cliente abrir ventana de cliente e ir al paso 7.
7. El sistema muestra menú e ir al paso 9.
8. El sistema muestra pestaña de Administrador e ir al paso 8.
9. El cliente o el administrador pulsan el botón de cambiar tema.
10. El sistema muestra las opciones de temas.
11. El cliente o el administrador elijen tema.
12. El sistema cambia el tema de la pestaña al tema seleccionado

POSTCONDICIONES:

El cliente o el administrador han cambiado el tema.

NOMBRE: Insertar Productos

ID:CU1

DESCRIPCIÓN: El administrador solicita insertar un producto en el sistema. El sistema inserta los datos del producto introducido por el administrativo.

ACTORES: Administrador

PRECONDICIONES: Se requiere que el administrador este registrado en el sistema.

CURSO NORMAL:

1. El administrador solicita insertar producto.
2. El sistema solicita los datos del producto.
3. El administrador rellena el formulario con los datos
4. El sistema valida los datos introducidos.
5. El sistema comprueba que el nombre introducido para el producto nuevo no existe en el sistema y que ningún otro campo imprescindible este vacío.

6. El sistema muestra un mensaje de conformidad por pantalla, almacenando los datos introducidos.

POSTCONDICIONES:

Los datos introducidos del nuevo producto serán almacenados en un registro de la base de datos.

NOMBRE: Actualizar Productos

ID:CU2

DESCRIPCIÓN:

El administrador introduce los datos nuevos que quiera modificar de un producto seleccionado entre los existentes en el sistema.

ACTORES: Administrador

PRECONDICIONES:

Se requiere el gerente debe haberse identificado en el sistema.

y se debe hacer una búsqueda del producto al que nos queremos referir.

CURSO NORMAL:

1. El administrador solicita actualizar producto.
2. El sistema solicita los datos nuevos del producto.
3. El administrador rellena el formulario con los datos.
4. El sistema valida los datos introducidos.
5. El sistema muestra un mensaje de conformidad por pantalla, almacenando los datos introducidos.

POSTCONDICIONES: Los datos modificados del producto serán guardados en el mismo registro de la base de datos.

NOMBRE: Eliminar Productos

ID:CU4

DESCRIPCIÓN:

El administrador selecciona un producto y lo elimina del sistema.

ACTORES: Administrador.

PRECONDICIONES:

Se requiere el gerente debe haberse identificado en el sistema.

y se debe hacer una búsqueda del producto al que nos queremos referir.

CURSO NORMAL:

1. El administrador solicita eliminar producto.
2. El sistema muestra un mensaje de conformidad por pantalla, eliminando el producto.

POSTCONDICIONES:

El registro con los datos del producto que se quiere eliminar será suprimido de la base de datos.

NOMBRE: Realizar Backup

ID:CU5

DESCRIPCIÓN: EL administrador realiza backup

ACTORES: Administrador

PRECONDICIONES:

Se requiere el gerente debe haberse identificado en el sistema e iniciado sesión.

CURSO NORMAL:

1. El administrador pulsa el botón para realizar backup.
2. El sistema guarda los datos.

POSTCONDICIONES:

El administrador observa las ventas.

NOMBRE: Administrador Restaurar

ID:CU6

DESCRIPCIÓN: EL administrador restaurara los datos

PRECONDICIONES:

Se requiere que el administrador haya iniciado sesión y este modificando un producto

CURSO NORMAL:

- 1.El Administrador pulsa el botón restaurar
2. El sistema quita todos los cambios y devuelve a la condición original de los productos

POSTCONDICIONES:

El producto no recibió ningún cambio

NOMBRE: Comprar Productos

ID:CU1

DESCRIPCIÓN: El cliente podrá comprar los productos.

ACTORES: Cliente

PRECONDICIONES: Se requiere que el cliente este registrado en el sistema.

CURSO NORMAL:

- 1.El sistema muestra pestaña iniciar sesión
- 2.El cliente rellena los datos.
- 3.El sistema muestra el menú
1. El cliente selecciona añadir producto.
2. El cliente indica la cantidad de producto.
3. E El sistema valida la cantidad.
4. El sistema muestra pestaña del importe y el total.
5. El cliente pulsa el botón Imprimir

POSTCONDICIONES:

El cliente realiza la compra de productos.

NOMBRE: Eliminar Productos del Carro

ID:CU2

DESCRIPCIÓN: El cliente podrá cancelar los pedidos.

ACTORES: Cliente

PRECONDICIONES: Se requiere que el cliente haya comprado.

CURSO NORMAL:

1. El cliente selecciona el producto que desea eliminar del carro
2. El cliente selecciona el botón Eliminar

POSTCONDICIONES: El cliente realiza la cancelación de pedidos.

NOMBRE: Ordenar Productos

ID:CU3

DESCRIPCIÓN: El cliente ordena Productos.

ACTORES: Cliente

PRECONDICIONES: Se requiere que el cliente haya iniciado sesión.

CURSO NORMAL:

1.EL Cliente pulsa cualquier botón de modo que ordene como elija el Cliente de la tabla del menú.

1.1. EL cliente pulsa Productos

1.2. EL cliente pulsa Cantidad.

1.3. EL cliente pulsa id.

1.4. EL cliente pulsa precio.

2. El sistema del modo que haya elegido el cliente ordena.

POSTCONDICIONES:

El cliente o el administrador realiza la acción deseada.

NOMBRE: Factura

ID:CU4

DESCRIPCIÓN: El sistema realiza la factura

ACTORES: Cliente

PRECONDICIONES: Se requiere que el cliente haya iniciado sesión

CURSO NORMAL:

1. El cliente añade productos al carro
2. El cliente pulsa el botón de mostrar el carro
3. El sistema muestra el carro
4. El cliente da al botón terminar
5. El sistema muestra un mensaje del total y de imprimir
 - 5.1. El cliente pulsa el botón imprimir e ir paso 6.
 - 5.2. El cliente pulsa el botón cancelar.
6. El sistema imprime factura.
7. El sistema quita la pestaña de imprimir factura.

POSTCONDICIONES:

El Cliente ha visto niega Imprimir factura.

DISEÑO

Diagrama de clases

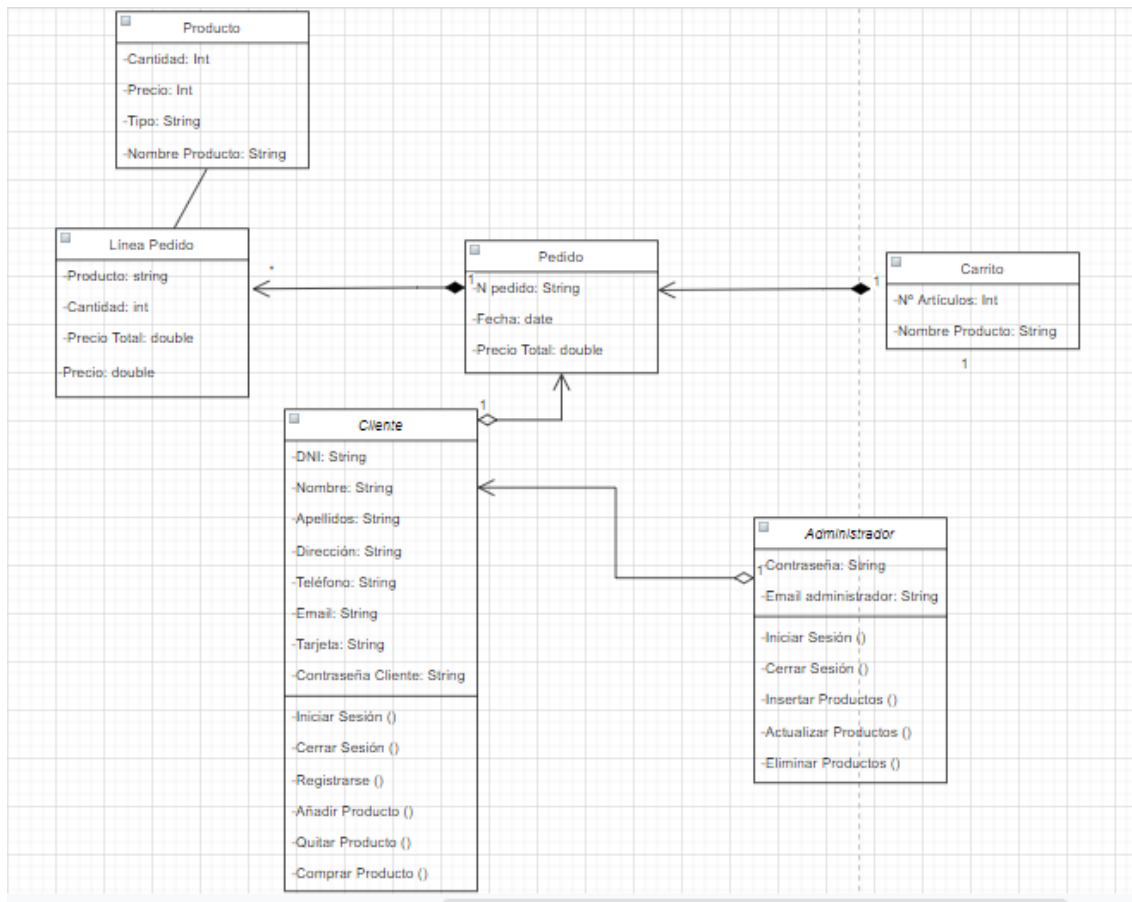
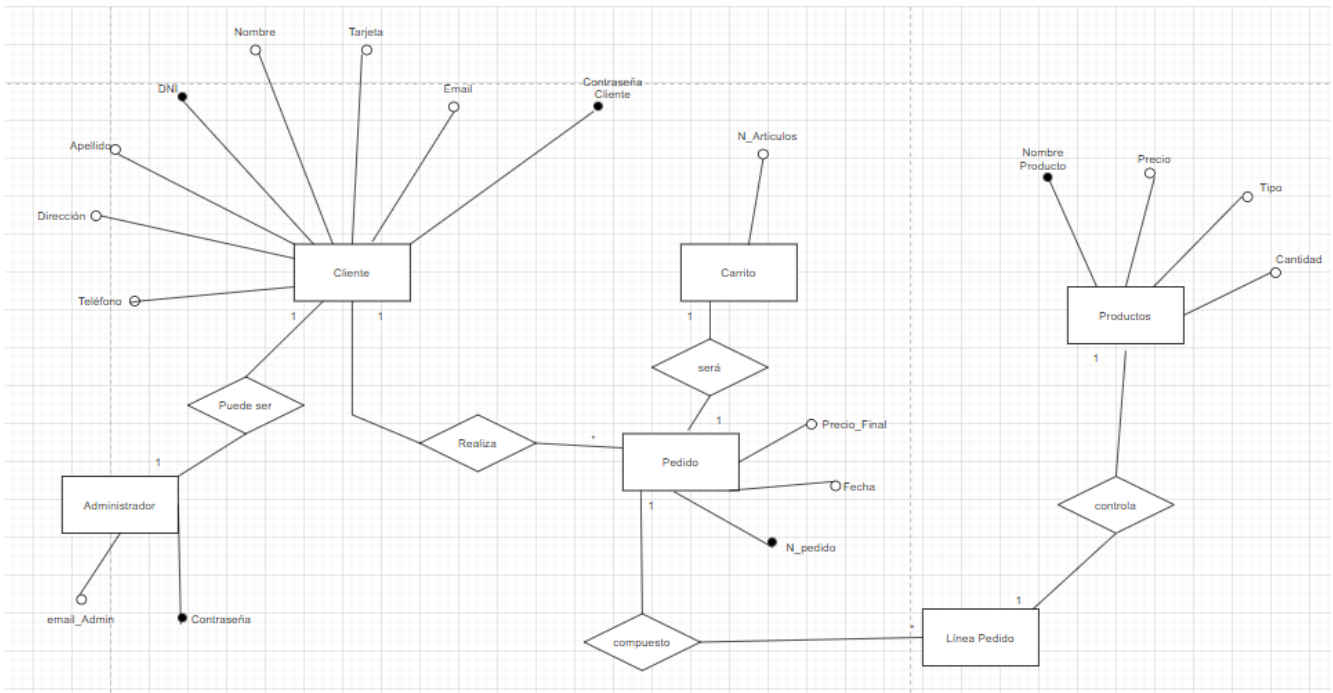


Diagrama de entidad/relación



Patrones

- **Fachada**: los usuarios sólo ven la vista, no ven todos los procesos que van ocurriendo por detrás. Implementado con todas las interfaces. Utilizado en:
 - En gran parte del proyecto.

- Singleton: es un patrón de diseño creacional que nos permite asegurarnos de que una clase tenga una única instancia, a la vez que nos proporciona un punto de acceso global a dicha instancia
Utilizado en:
-Backup.
-Controladores.
-Repositorios.
-DataBaseManager.
- Inyección de dependencias: se encarga de extraer la responsabilidad de la creación de instancias de un componente para delegarla en otro
Utilizado en:
-Controladores.
-Repositorios.
- Patrón Observer: permite definir un mecanismo de suscripción para notificar a varios objetos sobre cualquier evento que le suceda al objeto que están observando.
Utilizado en:
-Modelos.
-Controladores.
-Repositorios.
-DataBaseManager.
- Principios SOLID: es un acrónimo acuñado por Robert C. Martin en el cual se representan los cinco principios básicos de la programación orientada a objetos. La intención de seguir estos principios es eliminar malos diseños, evitar la refactorización y construir un código más eficiente y fácil de mantener.
Utilizado en:
-En gran parte del proyecto.
- Modelo vista-controlador: es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Tareas ejercidas por cada componente del equipo

- **Álvaro Mingo Castillo**
 - Controladores
 - Repositorios
 - Interfaces: carritoView
 - Testing
 - Modelos
 - Diseño y planificación del desarrollo.
 - Documentación
 - Diagrama de casos de uso.
 - Diagrama de entidad/relación
 - Asignación del título del proyecto.

- Jefe de Proyecto: Jorge Sánchez Berrocoso

- Controladores

- Repositorios

- Interfaces

- Testing

- Modelos

- Diseño y planificación del desarrollo.

- Informe

- Casos de uso.

- Asignación del título del proyecto.

- Requisitos funcionales, no funcionales y de información.

- Backup

- Aportación de imágenes

- implementación HTML en java

- Iván Azagra Troya

- Controladores
- SceneManager
- Interfaces
- Bocetos de interfaces
- Diseño y planificación del desarrollo.
- Asignación del título del proyecto.
- Requisitos funcionales, no funcionales y de información.
- Backup
- Aportación de imágenes
- DataBaseManager
- Gestión de almacenamiento.
- Tratamiento de datos.
- Diseño gráfico.

- Marcos Funez

- Diagrama de clases.

COSTE DEL PROYECTO

El equipo de informáticos del IES Luis Vives que realizará el proyecto que será encargado por una gran empresa de supermercados. El equipo va a dedicar los recursos dados por la empresa a realizar el proyecto con la intención de gastar los mínimos recursos.

-Tiempo del proyecto: 2 meses

-Precio de recursos: 0 €

-Precio de licencias: 0 €

-Precio del Proyecto: 36 000 €

-El equipo informático estará compuesto por tres Java Junior Developer.

-Salario de cada Java Junior Developer: 1500 €

-Dinero recibido a cada Java Junior Developer: 3000 €

- Dinero Obtenido por el jefe de empresa: 27.000 €.



PROYECTO PROGRAMACIÓN DIADAM

