

Definiciones, Acrónimos, y Abreviaciones

Componente: un componente es una pieza de hardware que se vende dentro de la tienda, puede ser una tarjeta gráfica, un procesador o un módulo de RAM, por ejemplo.

Pedido: un pedido es el carrito comprado de un cliente el cual puede contener 1 o más computadoras, cada una con sus propios componentes específicos.

Incompatibilidad: hay determinados casos donde 2 componentes específicos no funcionan bien entre si aunque sean compatibles o no debido a alguna falla, versionado del sw o porque no pueden funcionar juntos, a eso se lo conoce como una incompatibilidad

RMA: se le llama RMA a cuando se detecta un error durante el armado o luego de la entrega dentro de la computadora, puede ser una falla de software como de hardware, así como una incompatibilidad.

Presupuesto: se le dice presupuesto a la cantidad de dinero del cliente que quiere gastar en una computadora.

Armado de computadora: el armado de una computadora se refiere al proceso durante el cual se toman los componentes separados y se los junta para posterior instalar el sistema operativo, de esta forma se culmina en una computadora ya armada.

Canales del CPU: cantidad de slots de memoria del CPU que deben ser ocupados para mejor rendimiento.

PSU: fuente de poder de una computadora que cuenta con un valor de watts para poder darle energía

Listado de casos de uso:

- Negocio
 - CU-NEG-001 Realizar venta
 - CU-NEG-002 Realizar compra
 - CU-NEG-003 Realizar armado del pedido
 - CU-NEG-004 Informar error de componente
 - CU-NEG-005 Entregar pedido
 - CU-NEG-006 Informar error del equipo
 - CU-NEG-007 Generar reporte
 - CU-NEG-008 Alta de componente
 - CU-NEG-009 Baja de componente
 - CU-NEG-010 Modificación de componente
 - CU-NEG-011 Registro de empleado
 - CU-NEG-012 Baja de empleado
 - CU-NEG-013 Modificación de datos del empleado
 - CU-NEG-014 Registro de cliente
- Técnicos
 - CU-TEC-001 Login
 - CU-TEC-002 Logout
 - CU-TEC-003 Cambiar contraseña
 - CU-TEC-004 Consultar a bitácora
 - CU-TEC-005 Gestionar roles y permisos de empleados
 - CU-TEC-006 Gestión multi idioma de la pagina

Identificación del caso de uso:

- CU-NEG-001

Nombre del caso de uso:

- Realizar venta

Descripción del caso de uso:

- El cliente compra una o más computadoras conformando un pedido basándose en su presupuesto en pesos argentinos y su tipo de uso, el sistema dará de alta el pedido correspondiente con estado pendiente
 - El sistema creara de forma automática las computadoras a mostrarle al cliente

Precondiciones

- El cliente tendrá que estar logueado en el sistema
- Se tendrá que contar con stock de componentes para armar al menos una computadora

Postcondición

- Se dará de alta el pedido con estado pendiente asignando al empleado más ocioso
- Se vera actualizado el stock de componentes
- Se dará de alta la factura correspondiente al pedido

Actores primarios

- Cliente

Actores secundarios

- ---

Disparador

- El cliente quiere comprar una computadora de manera automática.

Escenario principal de éxito

1. El usuario se acerca a la sección de “comprar computadora”
2. El sistema le da bienvenida y carga la pantalla “crear presupuesto” con un formulario a completar por el cliente con los siguientes campos:
 - A. **Presupuesto:** se hará a través de un caja de texto numérico donde se pueden colocar hasta 8 dígitos y además debe contar con flechas para subir o bajar la denominación
 - B. **Tipo de uso:** se cargarán los distintos tipos de uso de la tienda en formato string con un largo de 30 caracteres como máximo a través de un dropdownlist y se mostraran en el listado, ejemplos de esto pueden ser: “gaming” – “diseño gráfico” – “arquitectura”
 - C. **Importancia:** se cargarán 2 opciones a través de un dropdownlist, siendo estas precio o calidad donde el cliente indicara que es más importante para él, que la computadora sea mas barata o sea de mejor calidad
3. El cliente carga los datos de formulario con su presupuesto en pesos argentinos, su tipo de uso e importancia.
4. El sistema espera la confirmación del usuario
5. El usuario confirma haciendo click en “buscar computadoras” para continuar con el proceso de búsqueda de computadoras.
6. El sistema creara las 5 mejores computadoras basado en el tipo de uso del cliente, presupuesto e importancia.
 - A. Se toma el tipo de uso del cliente y se consulta la especificación en base a ese tipo de uso, aquí se guardan las directivas para el armado de la computadora en general, como pueden ser:
 - A. **Calidad CPU:** dato entero entre 1 y 10
 - B. **Calidad Mother:** dato entero entre 1 y 11 (debe ser autocalculada tomando la calidad del CPU + 1)
 - C. **Calidad FAN:** dato entero entre 1 y 10
 - D. **Capacidad RAM:** dato entero y múltiplo de 4
 - E. **Capacidad HDD:** dato entero y múltiplo de 500
 - F. **Capacidad SSD:** dato entero y múltiplo de 256
 - G. **Calidad Video:** dato entero entre 1 y 10
 - H. **Calidad tower:** dato entero entre 1 y 10
 - I. **Tipo de uso:** string que es usado para mostrárselo al usuario y luego usado para consultar a la especificación en particular dentro de la tabla

7. El sistema ordena todos los componentes con stock según la importancia del cliente (calidad o precio)
Cálculo de orden:
Calidad: Componentes.OrderByDescending(Componente. Calidad)
Precio: Componentes.OrderByDescending(Componente. Precio)
Calculo stock:
Stock: Componente. Stock >= 1
8. El sistema consulta todos los CPUs y los filtra por aquellos que son mayores o iguales en calidad a la especificación calidad CPU.
Calculo: CPU. Calidad >= especificación calidad CPU
9. El sistema toma el CPU valido actual y lo agrega a la computadora en armado.
10. El sistema consulta todas las Mother y las filtraran tomando en cuenta que tengan el mismo socket que el CPU y a su vez tengan una calidad mayor a la especificación calidad Mother
Calculo: Mother.Socket == CPU.Socket && mother.Calidad >= especificación Calidad Mother
11. El sistema toma la primera Mother y la agrega a la computadora en armado.
12. El sistema consulta todas las RAMs y las filtra tomando en cuenta que tengan el mismo tipo de memoria que el CPU y que tengan la capacidad suficiente para cumplir con la especificación divido la cantidad de canales del CPU, además valida que las velocidades máximas de CPU, Mother y RAM sean acordes.
Calculo: ram.TipoMemoria == cpu.TipoMemoria && ram.Capacidad >= (especificaciones.CapacidadRam / cpu.CantidadCanales) && ram.FrecuenciaMaxima <= cpu.FrecuenciaMaxima && ram.FrecuenciaMaxima <= mother.FrecuenciaMaxima
13. El sistema toma la primera RAM y en misma cantidad a los canales tenga el CPU para agregar a la computadora.
14. El sistema validara si se necesita una GPU para la computadora en armado ya que no se requiere en caso de que la calidad de video del CPU y una Mother con video integrada sea acorde a las especificaciones, por lo que se validara así:
Calculo: mother.TieneVideoIntegrado && cpu.CalidadVideo >= especificaciones.CalidadVideo
15. El sistema valida que hace falta una GPU por lo que consulta todas las GPUs y las filtra tomando en cuenta que su calidad sea mayor o igual a la calidad de video esperada:
Calculo: gpu.Calidad >= especificaciones.CalidadVideo
16. El sistema toma la primera GPU y la agrega a la computadora en armado.
17. El sistema validara si se necesita un FAN para la computadora en armado ya que no se requiere en caso de que el nivel del fan del CPU sea mayor o igual a la calidad esperada de la especificación.
Calculo: cpu.CalidadFan >= especificaciones.CalidadFan
18. El sistema valida que hace falta un FAN por lo que consulta todos los FANs y los filtra tomando en cuenta que su calidad sea mayor o igual a la calidad de video esperada y que además sea compatible con el CPU que se ha asignado.
Calculo: fan.CalidadFan >= especificaciones.CalidadFan && fan.Socket.Contains(cpu.Socket)
19. El sistema toma el primer FAN y lo agrega a la computadora en armado.

20. El sistema consulta todas las Towers y las filtra tomando en cuenta el tamaño de la Mother y del fan, así como de que la calidad sea mayor o igual a la calidad de tower esperada
Cálculo: `tower.TamanoFormato >= mother.TamanoFormato && tower. Calidad >= especificaciones.CalidadTower && tower.FanSize >= fan.Capacidad`
21. El sistema toma la primera Tower y lo agrega a la computadora en armado.
22. El sistema consulta todas las PSUs y las filtra tomando en cuenta el wattage total de la computadora en armado sea mayor o igual a la suma de consumo de todos los componentes sumado a un excedente del 30% (% de energía máxima que se perdería de la fuente)
Cálculo: `psu.Capacidad >= (Componentes. Sum(Consumo) * 1.30)`
23. El sistema toma la primera PSU y la agrega a la computadora en armado.
24. El sistema valida que el precio computadora es valido para el presupuesto, es decir que su precio sea menor o igual al presupuesto en pesos argentino.
Cálculo: `Componentes. Sum(Precio) <= presupuesto`
25. El sistema agrega la computadora a los armados a mostrar
26. El sistema finaliza su proceso de armado de computadoras mostrando la pantalla de “selección de computadora” donde se muestra en una grilla las mejores 5 computadoras con sus características principales junto con una descripción a la mejor computadora para el cliente
A. **Cálculo:** `computadoras.OrderBy(calidad).Top(5)`
B. **Cálculo:** `computadoras.OrderBy(precio).Top(5)`
27. El usuario agrega las computadoras que desea haciendo click en la grilla seleccionando la computadora con el botón de “agregar computadora”.
28. El sistema agrega las computadoras seleccionadas por el usuario a una lista debajo de la grilla y espera la confirmación del usuario.
A. Se validará al momento de agregar la computadora a lista el stock de cada uno de los componentes que conforman la computadora en base a su cantidad y luego se agrega a listado de computadoras requeridas.
Cálculo: `Componente. Stock >= cantidad requerida.`
29. El usuario confirma el pedido con las computadoras agregadas haciendo click en el botón de “finalizar compra”.

30. El sistema lleva al usuario a la pantalla de “finalizar compra” donde le muestra un formulario con los siguientes datos:

- A. **Método de pago:** se cargarán un dropdownlist con 2 elementos -> Transferencia o Efectivo
- B. **Envío:** se cargarán un dropdownlist con 2 elementos -> Envío por correo o Retiro en tienda
- C. **En caso de tener la opción seleccionada de envío por correo, se cargarán los siguientes campos:**
 - A. **Localidad:** se cargarán las localidades validas de envío para la tienda en un dropdownlist en formato de 30 caracteres.
 - B. **Código postal:** se cargará un textbox con formato alfanumérico y de 10 caracteres
 - C. **Calle:** cargará un textbox con formato alfanumérico y de 25 caracteres
 - D. **Altura:** se cargará un textbox numérico con formato valido para 5 números
 - E. **Entre calle y calle:** cargará un textbox con formato alfanumérico y de 100 caracteres

31. El usuario completa los datos del envío.

32. El sistema espera la confirmación del cliente

33. El cliente confirma la compra haciendo click en el botón “comprar”

34. El sistema actualiza el stock por cada una de las computadoras.

Calculo: se crea un diccionario o mapa formado por el componente y su cantidad, luego se toman todos las computadoras y seleccionan tantos componentes tenga, una vez que tenemos todos los componentes los iteramos, en caso de que el componente ya se encuentre en el diccionario se le sumara +1 a la cantidad en caso contrario se le asignara 1, de esta manera sabremos cuantos componentes hay en esas computadoras

Ejemplo de implementación

```
var componenteCantidad = new Dictionary<Componente, int>();
foreach (var componente in Computadoras.SelectMany(computadora => computadora.Componentes))
    componenteCantidad[componente] = componenteCantidad.ContainsKey(componente) ? +1 : 1;
```

35. El sistema asigna el pedido al empleado más ocioso al pedido para realizarlo

Calculo: se crea un diccionario o mapa formado por el empleado y el número de computadoras a armar, luego se toman todos los pedidos y se los itera, en caso de que el diccionario ya tenga ese empleado se le suman las computadoras del pedido, en caso contrario se le asigna esa cantidad de computadoras.

Por ultimo y una vez que ya tengamos todos pedidos iterados se hace una búsqueda por la Key con menor Value que es por lo tanto el empleado más ocioso.

Ejemplo de implementación:

```
var empleadoPedido = new Dictionary<Empleado, int>();
foreach (var pedido in ObtenerPedidos())
{
    if (empleadoPedido.ContainsKey(pedido.Empleado))
    {
        empleadoPedido[pedido.Empleado] += pedido.Computadoras.Count;
    }
    else
    {
        empleadoPedido[pedido.Empleado] = pedido.Computadoras.Count;
    }
}
var empleadoOcioso = empleadoPedido.First(empleado => empleado.Value == empleadoPedido.Values.Min()).Key;
```

- 36.** El sistema asigna el cliente logueado al pedido
- 37.** El sistema da de alta el pedido
- 38.** El sistema da de alta la factura y se la envía al cliente a través de mail
 - A.** Para ello se utilizará un cliente de email como puede ser SMTP client.
- 39.** El sistema notifica que se genero el pedido con éxito junto con el número de pedido
 - A.** El numero de pedido se calcula en base al ID con el que se subió el pedido.

Caminos alternativos

Para el paso 15

1. El sistema valida que la GPU no es necesaria.
2. Se sigue por el paso número 17 del flujo principal.

Para el paso 17

1. El sistema valida que el FAN no es necesario
2. Se sigue por el paso número 20 del flujo principal

Para el paso 24

1. El sistema valida que la computadora excede el presupuesto del cliente
2. Se sigue por el paso número 26 del flujo principal

Para el paso 25

1. Todavía hay algún CPU valido para poder armar otro armado
2. Se sigue por el paso número 9 del flujo principal

Para el paso 26

1. El sistema no encontró ninguna computadora acorde al tipo de uso y presupuesto
2. El sistema notifica al cliente que intente con un mayor presupuesto o con otro tipo de uso
3. Se sigue por el paso número 2 del flujo principal

Para el paso 28

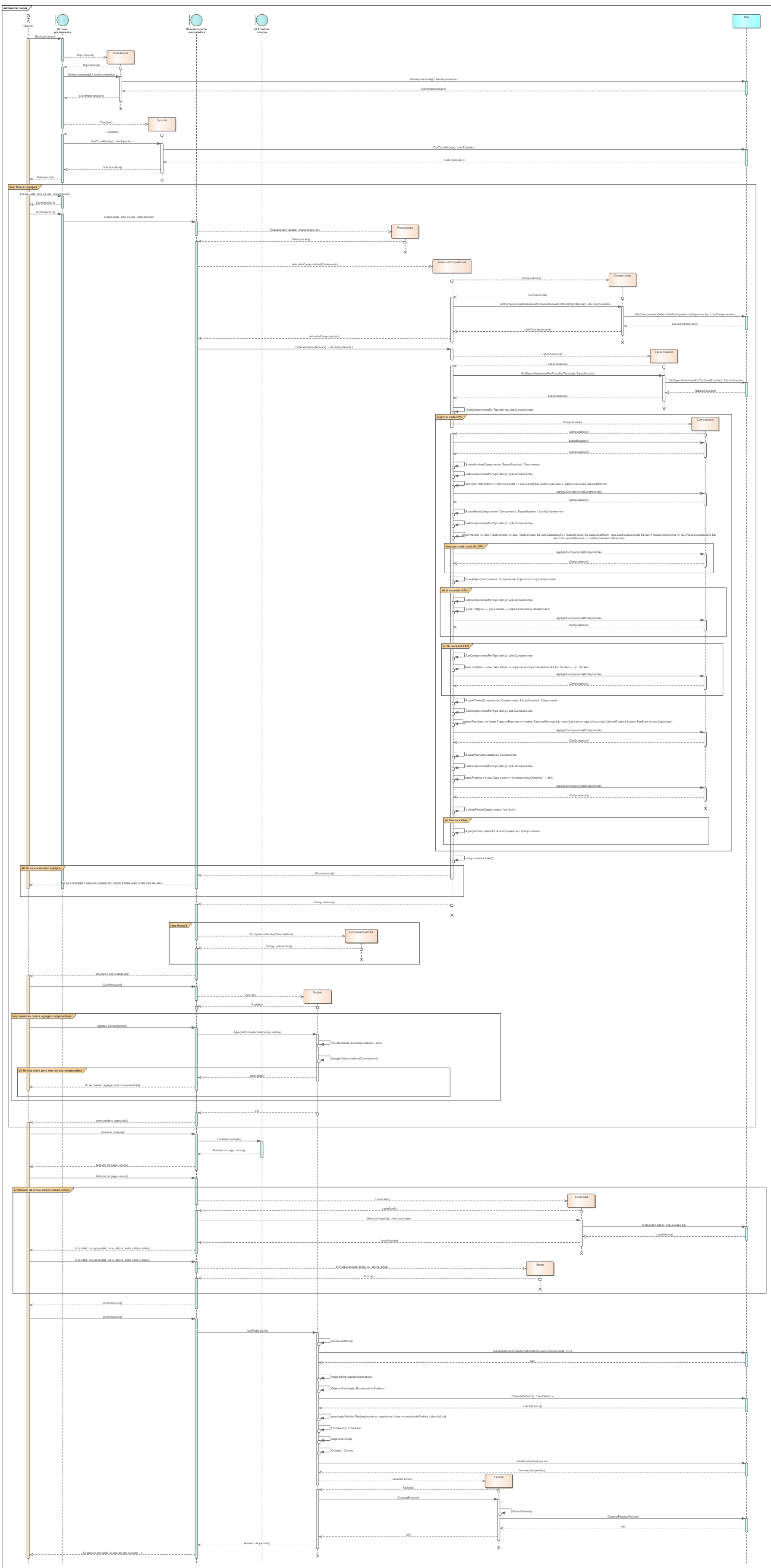
1. El sistema informa que no tiene mas stock para esa computadora
2. Siguiendo por el paso número 28 del flujo principal

Para el paso 29

1. El usuario quiere seguir agregando equipos con otro tipo de uso y presupuesto
2. Siguiendo por el paso número 2 del flujo principal

Para el paso 31

1. El usuario selecciona retiro en tienda en las opciones
2. Siguiendo por el paso numero 32 del flujo principal



Pantallas:

Crear presupuesto

FSolutions ECommerce

Comprar una computadora

Acerca de

Contacto

Hola! indicanos tu presupuesto y para que vas a utilizar la computadora!

Presupuesto: \$

Tipo de uso:

Gaming

Importancia:

Precio

Buscar computadoras

© 2022 - Mi aplicación ASP.NET

Selección de computadora

FSolutions ECommerce

Comprar una computadora

Acerca de

Contacto

Armamos estas computadoras para ti!

	Cpu	Ram	Almacenamiento	Gpu	Precio	Descripcion
<div>Agregar computadora</div>	Ryzen 5 3400G	12GB	1TB HDD 256 SSD	Integrada	80000	Esta es la mejor computadora para vos!
<div>Agregar computadora</div>	Ryzen 7 5600X	16GB	1TB HDD 256 SSD	GTX 1650TI	100000	---
<div>Agregar computadora</div>	Intel i7	16GB	1TB HDD 256 SSD	GTX 1650TI	110000	---
<div>Agregar computadora</div>	Intel i9	16GB	1TB HDD 256 SSD	GTX 1650TI	120000	---
<div>Agregar computadora</div>	Ryzen 9	16GB	1TB HDD 256 SSD	GTX 1650TI	125000	---

Computadoras agregadas:

cantidad = 1

- Ryzen 5 3400G 12GB 1TB HDD 256 SSD Integrada \$80000

cantidad = 1

- Ryzen 7 5600X 16GB 1TB HDD 256 SSD GTX 1650TI \$100000

Finalizar Compra

© 2022 - Mi aplicación ASP.NET

Finalizar compra

FSolutions ECommerce

[Comprar una computadora](#)

[Acerca de](#)

[Contacto](#)

Indique su metodo de pago y como quiere retirarlo y la computadora ya va a ser tuya!

Metodo de pago:

Transferencia ▼

Seleccionar envio:

Envio por correo ▼

Localidad: Buenos aires ▼

Codigo postal: 1653

Calle: Republica

Altura: 4877

Entre calle y calle: Alvear y Lacroze

Comprar

Se ha generado con exito el pedido con numero: 1234.

Identificación del caso de uso:

- CU-NEG-002

Nombre del caso de uso:

- Realizar compra

Descripción del caso de uso:

- Se realizan las compras de los componentes bajos en stock

Precondiciones

- El empleado tendrá que estar logueado en el sistema con permisos de empleado
- Se tendrá que contar con al menos un componente bajo de stock y un proveedor para ese componente

Postcondición

- Se actualiza el stock de los componentes que están bajos de stock
- Se da de alta la orden de compra, orden de pago y recibo de las compras realizadas
- Se actualiza el precio de los componentes en base a los nuevos ingresos

Actores primarios

- Empleado

Actores secundarios

- ---

Disparador

- El empleado detecta un stock bajo de componentes y quiere realizar un pedido a proveedores

Escenario principal de éxito

1. El sistema notifica que se cargaran los componentes con stock bajo.
2. El empleado confirma la notificación
3. El sistema carga los componentes que están bajos de stock al mejor precio en base a los distintos proveedores.

a. **Calculo stock bajo:** componente. Stock <= componente.LimiteStock

b. **Calculo precio más bajo:** se iterarán todos los componentes bajos en stock y se buscaran los mismos dentro los componentes en venta de cada uno de los proveedores, luego se validará si el proveedor actualmente iterado tiene un mejor precio y así sucesivamente con los demás.

Para el calculo de precio se efectuara lo siguiente ComponenteEncontrado.precio <= precio mínimo encontrado.

Luego de iterar todos los proveedores para ese componente se guardará dentro de un ítem de compras que proveedor se encontró el mas barato junto con el componente.

Una vez iterados todos los componentes nos quedaran los ítems de compra con el precio mas bajo basado en todos los proveedores de la tienda.

Ejemplo de implementación:

```
foreach (var componente in componentesBajoStock)
{
    decimal precioMinimo = 0;
    Proveedor proveedorMinimo = new Proveedor();
    foreach (var proveedor in proveedores)
    {
        var componenteEncontrado = proveedor.GetComponenteEnVentaPorId(componente.Id);
        if (componenteEncontrado.Precio <= precioMinimo)
        {
            precioMinimo = componenteEncontrado.Precio;
            proveedorMinimo = proveedor;
        }
    }
    componente.Precio = precioMinimo;
    AgregarItemCompra(compras, componente, proveedorMinimo);
}
```

4. El sistema muestra una la pantalla de “Revisión de stock” con una grilla que muestra los componentes bajos en stock junto con el mejor precio conseguido seguido del nombre del proveedor y un botón para poder agregar a las compras junto con un textbox para solicitar la cantidad a comprar, junto con los siguientes campos:
 - a. **Cantidad:** textbox numérico con máximo de 2 números y que cuentan con flechas para cambiar su denominación.
5. El sistema solicita ingresar la cantidad
6. El empleado ingresa la cantidad
7. El sistema solicita el componente
8. El empleado selecciona el componente que quiere agregar haciendo click en “agregar a compras”
9. El sistema espera la confirmación de la carga
10. El empleado hace click en “iniciar compra” confirmando la compra

11. El sistema da de alta las compras en estado “esperando aprobación” subiendo cada compra entre los distintos proveedores obtenidos
 - a. **Calculo:** se toman todos los ítems de la compra y se los agrupa por el ID del proveedor, luego por cada proveedor se sube los componentes comprados correspondiente
12. El sistema notifica que se cargo la compra con éxito seguido del ID de la compra creada
13. El empleado se dirige hacia la sección de aprobación de compras
14. El sistema carga la pantalla “Aprobación compras” con una grilla que muestra todas las compras sin aprobar, junto con un botón para poder aprobarlas
 - a. **Calculo:** Compras.FindAll(c => c.estado != aprobada)
15. El empleado confirma la aprobación de la compra que quiere aprobar haciendo click en el botón “aprobar”
16. El sistema pregunta al empleado si está seguro de aprobarla
17. El empleado confirma la aprobación compra
18. El sistema notifica que la compra ha sido aprobada
19. El sistema confirma la aprobación de la orden de compra y actualizando su estado a “aprobada”
20. El sistema envía la orden de compra al proveedor en cuestión mediante mail
 - a. Se utilizará un cliente de mail como SMTP client
21. El sistema actualizara los precios de los componentes
 - a. **Operación:** Componente. Precio = nuevo precio
22. El empleado espera a que la compra sea entregada por el proveedor
23. El proveedor trae los componentes de la compra hacia la tienda
24. El empleado se acerca a la sección de recibo de compras y hace click en “recibir compras”
25. El sistema carga la pantalla de “recibo compras” con una grilla mostrando los datos de las compras aprobadas y un botón para seleccionarla, junto con los siguientes campos:
 - a. **Id Compra:** se autocompletará con el Id de la compra seleccionada por lo que será readonly por parte del usuario
 - b. **Componente:** dropdownlist que se completaran con los componentes pedidos para esa compra
 - c. **Cantidad recibida:** textbox numérico que indique la cantidad recibida de ese componente seleccionado
 - d. **Recibo completo:** checkbox que indica si se recibió todo la cantidad de ese componente seleccionado
 - i. Si se mantiene el checkbox el campo de cantidad recibida se debe de invalidar.

Calculo: Compras.FindAll(c => c.estado == aprobada)

26. El empleado selecciona el componente a cargar
27. El sistema solicita la cantidad recibida

28. El empleado carga la cantidad recibida del componente seleccionado
29. El sistema agrega a la lista de componentes recibidos el componente cargado junto con su cantidad
30. Se repite el paso del flujo principal 26 hasta que el empleado termine la carga
31. El sistema espera la confirmación del empleado para la carga
32. El empleado confirma la carga de los componentes haciendo click en “generar recibo y orden de pago”
33. El sistema verifica si la compra ha sido recibida de manera total o parcial
 - a. **Calculo:** por cada componente recibido se verifica lo siguiente:
`Compra.ComponentesParaComprar.Count(c => c.Id == Id) == cantidad recibida`
34. El sistema verifica que se ha recibido la compra de manera completa
35. El sistema actualiza el stock de los componentes comprados
36. El sistema actualiza el estado de la compra a “Completa”
37. El sistema da de alta una orden de pago hacia el sector de cobranzas
38. El sistema da de alta el recibo asociado a la compra
39. El sistema envía el recibo de la compra hacia el proveedor mediante mail
 - a. Se utilizará un cliente de mail como SMTP client
40. El sistema notifica que se ha cargado con éxito el recibo y la orden de pago
41. El empleado se acerca a la sección de órdenes de pago y hace click en “pagar órdenes de pago”
42. El sistema notifica que se cargara las órdenes de pago impagas
43. El sistema carga la pantalla de “Pagar órdenes de pago” con una grilla mostrando las órdenes de pago impagas junto con los datos de los proveedores y el id de la compra junto con un botón llamado “marcar como pagada” para poder pagarla
 - a. **Calculo:** `ObtenerOrdenesDePago().Where(orden => !orden.EstaPagada)`
44. El sistema espera la confirmación del pago por parte empleado
45. El empleado selección y confirma la orden de pago haciendo click en “marcar como pagada” dentro de la compra pagada
46. El sistema notifica que se marcó como pagada la orden de pago
47. El sistema actualiza la compra hacia el estado “pagada”
48. El sistema notifica mediante mail al proveedor que se ha pagado con éxito la orden de pago.
 - a. Se utilizará un cliente de mail como SMTP client

Caminos alternativos

Para el paso 10

1. El empleado quiere seguir agregando componentes a la compra
2. Se sigue por el paso número 4 del flujo principal

Para el paso 15

1. El empleado no encuentra la compra aprobada
2. Se sigue por el paso número 3 del flujo principal

Para el paso 34

1. El sistema verifica que la compra se ha recibido de manera parcial
2. El sistema actualiza el stock de los componentes comprados
3. El sistema actualiza el estado de la compra a "Parcialmente completa"
4. El sistema da de alta el recibo asociado a la compra
5. El sistema envía el recibo de la compra hacia el proveedor mediante mail
 - a. Se utilizará un cliente de mail como SMTP client
6. Se sigue por el paso numero 17 del flujo principal

Pantallas:

Revisión de stock

Tenemos estos componentes con bajo stock

	Nombre	Precio	Tipo	Socket	StockBajo	Proveedor
Agregar a compras	Ryzen 5 3400G	10000	cpu	AM4	2	Juan SRL
Agregar a compras	Ryzen 7	10000	cpu	AM4	2	Import test SA
Agregar a compras	Intel i7 11th	10000	cpu	AM4	2	Import test SA

Cantidad:

Listado de componentes a comprar:

- Cantidad 2 - Ryzen 5 3400G 10000 cpu

Iniciar compra

Compra aprobada con ID 3!

Aprobación de compras

Listado de compras pendientes de aprobacion:

	Id	Proveedor	PrecioAPagar	Estado	EstaPagada
Aprobar	1	Jose SRL	1000000	En revision	<input type="checkbox"/>
Aprobar	2	Import test SA	1000000	En revision	<input type="checkbox"/>
Aprobar	3	Pepe gpus SRL	1000000	En revision	<input type="checkbox"/>

Recibo compras

Recibo Compras

Seleccionar	Detalle	Id Proveedor	PrecioAPagar	Estado	EstaPagada
Seleccionar	Detalle	1 Jose SRL	1000000	Aprobada	<input type="checkbox"/>
Seleccionar	Detalle	2 Import test SA	1000000	Aprobada	<input type="checkbox"/>
Seleccionar	Detalle	3 Pepe gpus SRL	1000000	Aprobada	<input type="checkbox"/>

Id compra:

1

Componente:

Ryzen 5 3400 G

Cantidad recibida:

3

☐ Recibo completo

Agregar a componentes recibidos

Componentes recibidos:

Cantidad: 3 - Ryzen 5 3600G

Generar recibo y orden de pago

Pago órdenes de pago

Ordenes de pago sin pagar

	Proveedor	TotalAPagar	EstaPagada	IdCompra
Marcar como pagada	jose SRL	10000	<input type="checkbox"/>	1
Marcar como pagada	marquitos SRL	10000	<input type="checkbox"/>	2
Marcar como pagada	susana SRL	10000	<input type="checkbox"/>	3

Diagrama de clases (Aclaración: todas las clases usan a la clase de acceso a datos -> DAL)

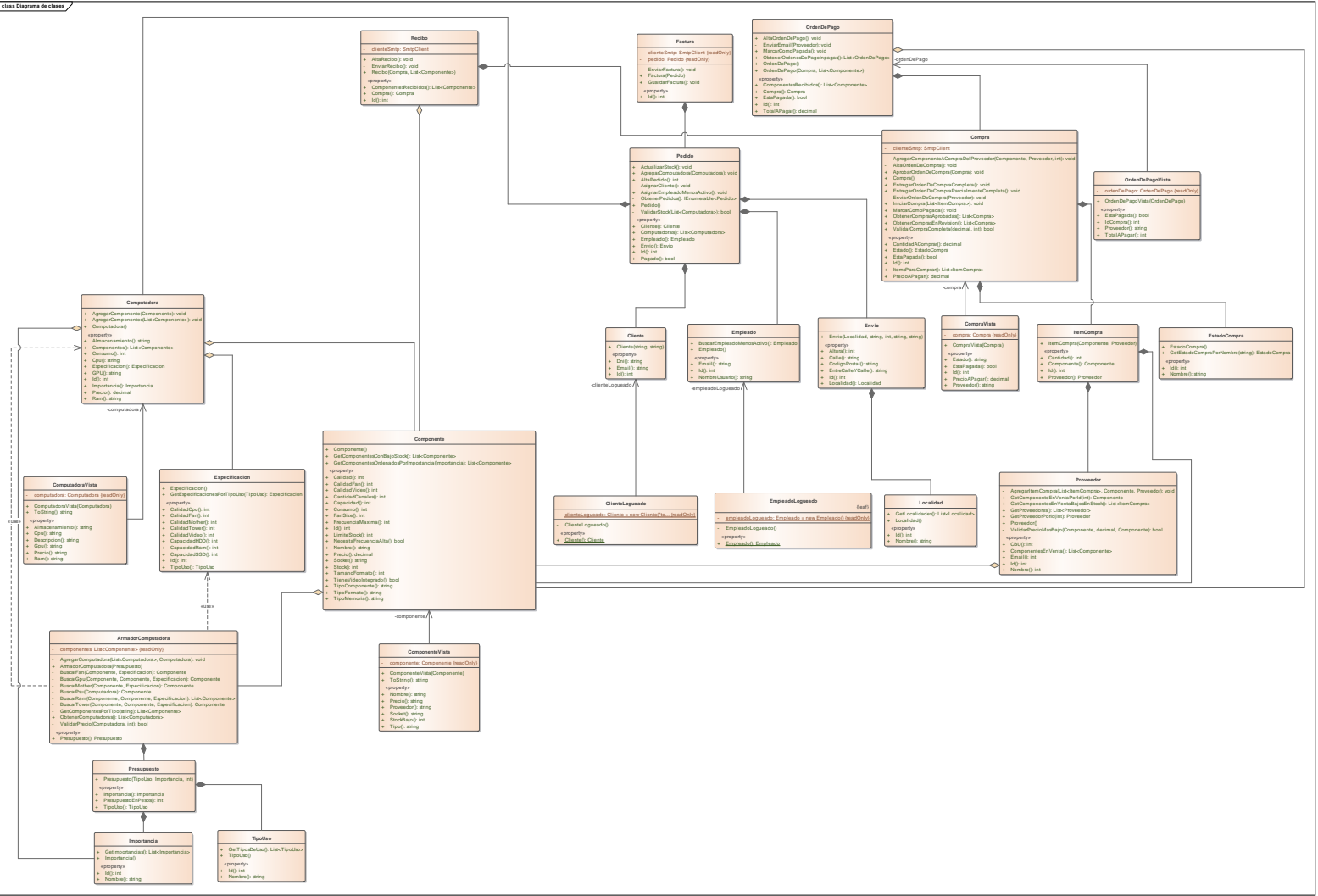


Diagrama entidad relación

dm DER

