=

Autor: Zoel Villar

Carrera: Ingeniería en sistemas

Comisión: 3A

Turno: Noche

Localización: Centro

**| APUNTES |**

**Trabajo de**

**Diploma**



Año 2023

Zoel Ivan Villar

Contents

[CARPETA DE PROYECTO 5](#_Toc147737229)

[G00. Descripción Global del Producto 5](#_Toc147737230)

[G01. Propósito 5](#_Toc147737231)

[G02. Descripción funcional del producto y Alcance 5](#_Toc147737232)

[PN1 Gestión de Ventas: 6](#_Toc147737233)

[PN2 Gestion de Compras: 7](#_Toc147737234)

[G03. Definiciones, Acrónimos, y Abreviaciones 7](#_Toc147737235)

[G04. Descripción de las personas participantes en el desarrollo del sistema de información y los usuarios (Roles) 9](#_Toc147737236)

[G05. Otros Requisitos 9](#_Toc147737237)

[G06 Diagramas de Clases y DER General 12](#_Toc147737238)

[G06.1 Diagrama de Clases general 12](#_Toc147737239)

[G06.3 Diagrama de Clases general Por Capas 13](#_Toc147737240)

[G07. DER general tipo Martin 14](#_Toc147737241)

[N00. Procesos de negocio 15](#_Toc147737242)

[N01. Especificación funcional por proceso de negocio 15](#_Toc147737243)

[N02. Especificaciones de Casos de Uso 20](#_Toc147737244)

[RFN2 – gestión de compras: 32](#_Toc147737245)

[T00. Documentos de aspectos técnicos que provee el sistema de información. 42](#_Toc147737246)

[T01. Arquitectura Base 42](#_Toc147737247)

[Mapa de Navegación 46](#_Toc147737248)

[T02.1 Gestión de usuarios. Crear 47](#_Toc147737249)

[T02.2 Gestión de Log In 51](#_Toc147737250)

[T02.3 Desbloquear Usuario 58](#_Toc147737251)

[T02.4 Cambiar Contraseña 62](#_Toc147737252)

[T02.5 Gestión de Logout 66](#_Toc147737253)

[T03. Gestión de Encriptado 70](#_Toc147737254)

[T03.1 Encriptacion Irreversible 70](#_Toc147737255)

[T03.2 Encriptación Reversible 71](#_Toc147737256)

[T04. Gestión de Perfiles de Usuario 72](#_Toc147737257)

[ASIGNAR PERFIL A USUARIO 77](#_Toc147737258)

[T05. Gestión de Múltiples Idiomas 81](#_Toc147737259)

[T06A Gestion de Bitacora de Evento 88](#_Toc147737260)

[Registro de Eventos – Dimension Interna 89](#_Toc147737261)

[Auditar bitácora de Eventos – Dimensión Externa 93](#_Toc147737262)

[T06b. Bitacora de cambios 97](#_Toc147737263)

[Guardar bitácora de Cambios –Dimension Interna 100](#_Toc147737264)

[Auditar bitácora de Cambios – Dimension Externa 102](#_Toc147737268)

[T07 GESTION DE RESPALDOS 106](#_Toc147737269)

[Gestion de respaldos - Backup 106](#_Toc147737270)

[Gestion de respaldos – Restore 109](#_Toc147737271)

[T08. Gestion de dígitos verificadores 112](#_Toc147737272)

[Generar Digito Verificador 115](#_Toc147737273)

[Comprobar Digito verificador 118](#_Toc147737274)

[Reparar inconsistencias en la base de datos 120](#_Toc147737275)

# CARPETA DE PROYECTO

# G00. Descripción Global del Producto

En este grupo se describe el producto con un conjunto de características de alto nivel que permiten comprender tanto el negocio como el producto de forma global.

# G01. Propósito

Una cafetería que recientemente abrió sus puertas en Lanús, Buenos Aires, llamada “Dialect Café”. Se pone en contacto con Ingenieros en Sistemas de UAI para desarrollar un sistema de compra y ventas que funcionará en su sede de manera presencial.

Se debe dejar de utilizar cualquier elemento físico, como libretas para llevar el día a día de las ventas, stock y totales.

En primer lugar, el sistema permitirá automatizar y optimizar los procesos de venta, lo que se traducirá en un ahorro de tiempo y esfuerzo para los empleados de la cafetería, que podrán centrarse en la atención al cliente en lugar de tener que preocuparse por llevar registros manuales de las ventas.

Además, el sistema permitirá mantener un control actualizado del inventario, lo que evitará que los empleados vendan productos que ya no están disponibles y permitirá realizar pedidos automatizados para reponer los productos que se estén agotando. De esta manera, se reducirá la pérdida de ventas y se mejorará la gestión de los recursos.

Por último, el sistema permitirá al gerente de la cafetería tener una visión clara y detallada de la situación del negocio, lo que le permitirá tomar decisiones informadas y estratégicas para mejorar la rentabilidad y la eficiencia de la cafetería.

En resumen, el sistema de gestión de compras y ventas será una herramienta valiosa para el éxito y la sostenibilidad de Dialecto Café.

# G02. Descripción funcional del producto y Alcance

RFN1 Gestión de ventas: El sistema permitirá a los empleados de la cafetería realizar ventas de productos a los clientes. Esto incluirá la capacidad de procesar pagos en efectivo y generar recibos de ventas. El recepcionista al completar el pedido del cliente, se le informará si hay disponibilidad de los productos o no, si los hay, se procede a procesar el pago y gestionar los productos pedidos por el cliente

## PN1 Gestión de Ventas:

1. El recepcionista recibe el pedido del cliente, donde se comunica el nombre, tipo y cantidad de los productos a consumir. **(Campos: Nombre del producto, tipo de producto, cantidad)**
2. Posteriormente, el recepcionista verifica la disponibilidad de los productos en el sistema. Si no hay disponibilidad, el cliente deberá tomar otra decisión, donde será asesorado por el recepcionista. **(Campos: Disponibilidad del producto)**
3. Si los productos están disponibles, el recepcionista ingresa la cantidad de productos que el cliente desea comprar en el sistema, incluyendo el nombre, tipo y cantidad. **(Campos: Nombre del producto, tipo de producto, cantidad)**
4. El sistema calcula el precio total de la compra y muestra el monto al recepcionista, que se comunica al cliente. **(Campos: Precio total)**
5. El cliente tiene la opción de realizar el pago, ya sea en tarjeta, MercadoPago o efectivo. En algunos de estos casos, se deben tomar en cuenta datos tales como banco, tipo de tarjeta, número de tarjeta, fecha de expiración de la tarjeta, número de cuenta, alias, monto a cancelar, DNI, apellidos, nombre, etc. **(Campos: Tipo de pago, datos de tarjeta, datos personales)**
6. El recepcionista ingresa el monto recibido del cliente en el sistema. **(Campos: Monto recibido)**
7. El sistema calcula el cambio y lo muestra al cliente. **(Campos: Cambio)**
8. El sistema genera dos recibos de venta que incluyen la lista de productos comprados, el precio unitario, la cantidad, el precio total de la venta, el monto recibido, el cambio y la fecha y hora de la transacción. Uno se lo entrega al cliente y otro se almacena para su control. **(Campos: Detalles de la venta, precios, cantidades, fechas)**
9. El sistema actualiza el inventario para reflejar los productos vendidos y la cantidad restante. **(Campos: Actualización del inventario)**

RFN2 Gestión de compras: El sistema permitirá al personal de la cafetería realizar pedidos de compra de productos a los proveedores. Esto incluirá la capacidad de generar órdenes de compra, realizar seguimiento del estado de los pedidos y recibir confirmaciones de los proveedores. También involucra la selección y registro de proveedores, la búsqueda de productos sin stock y el pago de la mercaderia.

## PN2 Gestion de Compras:

RFN2 – gestión de compras:

El Encargado de Dialect Café inicia el proceso de compras al analizar la existencia de productos en el inventario. El sistema le muestra una lista de productos que están bajos en stock o agotados, permitiendo al administrador identificar los productos que necesitan reponerse. El administrador selecciona los productos deseados y contacta al proveedor correspondiente para solicitar una cotización. Una vez que el proveedor responde con los precios y detalles de los productos, el sistema recibe y almacena la cotización para su revisión. Tras verificar los precios y términos, el administrador procede a generar una orden de compra, la cual se envía al proveedor para confirmar la adquisición. Una vez que los productos llegan a la cafetería, el administrador registra la recepción de productos en el sistema, realizando las actualizaciones necesarias en el inventario. Posteriormente, el sistema genera automáticamente una factura detallada para el proveedor. Este proceso garantiza un flujo eficiente y organizado para mantener el inventario actualizado y gestionar las compras de manera efectiva.

## Descomposición funcional

1. El sistema muestra una pantalla que enumera los productos con bajo stock o agotados, basándose en los niveles de inventario predefinidos. **(Campos: Nombre del producto, categoría, stock actual)**
2. El encargado de compras revisa la lista de productos y selecciona aquellos que necesitan reponerse.
3. Para cada producto seleccionado, el usuario ingresa la cantidad deseada para la compra. **(Campos: Cantidad a comprar)**
4. Una vez que se especifica la cantidad, se genera un documento de cotización. **(Campos: Proveedor, producto, cantidad, términos)**
5. El encargado de compras envía la solicitud de cotización a los proveedores correspondientes, incluyendo detalles como los productos solicitados y las cantidades necesarias.
6. Los proveedores reciben la solicitud de cotización y responden con precios y detalles de los productos. **(Campos: Precio, disponibilidad, tiempo de entrega)**
7. El administrador revisa las cotizaciones recibidas y compara los precios y términos ofrecidos por los proveedores. **(Campos: Precio, términos, disponibilidad)**
8. Si se aprueba una compra, el administrador genera una orden de compra en el sistema, especificando los productos, las cantidades y los proveedores seleccionados. **(Campos: Producto, cantidad, proveedor)**
9. El Encargado de compras envía la orden de compra al proveedor correspondiente para su confirmación.
10. El proveedor recibe la orden de compra y confirma la disponibilidad de los productos y los términos de la transacción. Se envían los productos
11. Cuando los productos llegan a la cafetería, el administrador registra la recepción de los productos en el sistema, ingresa el precio de venta y realiza las modificaciones necesarias, actualizando automáticamente el inventario. **(Campos: Producto recibido, precio de venta, modificaciones en inventario)**
12. El sistema genera una factura para el proveedor, que incluye los productos entregados, los precios acordados y los términos de pago. **(Campos: Productos entregados, precios, términos de pago)**
13. El administrador realiza el pago al proveedor. **(Campos: Confirmación de pago, fecha de pago)**

# G03. Definiciones, Acrónimos, y Abreviaciones

Aquí se detallan todas aquellas palabras, acrónimos y abreviaturas que poseen un sentido especial para el sistema o tienen un significado particular. Por ejemplo:

**Definiciones**:

CLIENTE: Persona física o jurídica que por lo menos ha adquirido una mercadería de la organización.

Proveedor: Persona o empresa que suministra productos o materiales a la cafetería.

PRODUCTO TERMINADO: Es un producto producido por la empresa disponible para la venta que puede estar constituido por materias primas y productos semielaborados.

CAFÉ: El café es una bebida aromática y estimulante que se prepara a partir de granos tostados seleccionados por su calidad y molidos de la planta del café. Estos granos pueden ser de diferentes nacionalidades, cada una con características y sabores distintos

Menú: Lista de productos o servicios que se ofrecen en la cafetería.

Ingredientes: Materias primas necesarias para la elaboración de los productos que se ofrecen en la cafetería.

Inventario: Lista detallada y actualizada de todos los productos o materiales que se tienen en la cafetería.

Factura: Documento legal que detalla los productos o servicios adquiridos y su costo.

Ticket: Recibo o comprobante de compra que se entrega al cliente al realizar una transacción.

**Acrónimos**:

POS: Point of Sale o Punto de Venta. Este término se refiere a la ubicación física en la que se realiza una transacción de venta, como puede ser una caja registradora, un mostrador o una mesa con un dispositivo de pago.

SKU: Stock Keeping Unit o Unidad de Mantenimiento de Existencias. Este acrónimo se utiliza para referirse a una identificación numérica o alfanumérica única asignada a un producto en inventario, lo que facilita su seguimiento y control.

FIFO: Primero en entrar, primero en salir (para el inventario de productos).

KPI: Indicadores clave de rendimiento.

ABM: Alta, baja y modificación.

PS: Permiso Simple

PC: Permico Compuesto

**Abreviaturas**:

ARG=Argentina

# G04. Descripción de las personas participantes en el desarrollo del sistema de información y los usuarios (Roles)

Roles que intervienen en Dialect Cafe

|  |  |  |
| --- | --- | --- |
| Zoel Villar | Analista en Sistemas, desarrollador del software | Creación de la arquitectura, alcance, desarrollo y mantenimiento del software |
| Federico Giubis | Administrador de Base de Datos | Administra la base de datos, creación, mantenimiento y seguridad de los datos. |
| Dante Richetti | QA y Testing | Pruebas del software antes de desplegarlo. |

|  |  |  |
| --- | --- | --- |
| Zoel Villar | Gerente | RFN1 – RFN2 |
| Lautaro Rojas | Administrador de sistemas | Todos los casos de uso |
| Guadalupe Almada | Encargado de compras | RFN2 |
| Mariana Ales | Encargado de Ventas / Recepcionista | RFN1 |

# G05. Otros Requisitos

Estandares de aprobación:

* El sistema debe ser fácil de usar e intuitivo para los usuarios. Debe haber una interfaz clara y organizada, que permita navegar de manera sencilla por las diferentes funcionalidades del sistema.
* Debe haber una buena capacidad de respuesta y estabilidad en el sistema, evitando errores o fallos en el funcionamiento que puedan afectar la experiencia de usuario o la efectividad del negocio.
* La seguridad de la información es crítica. Por lo tanto, el sistema debe garantizar la confidencialidad, integridad y disponibilidad de los datos de la cafetería y sus clientes.
* El sistema debe tener documentación y soporte técnico adecuados, para que los usuarios puedan solucionar problemas y resolver dudas de manera rápida y efectiva.
* El logo de la empresa debe ser visible en el programa en el Login y pantalla Principal
* El tiempo de respuesta del login, la carga de formularios y la carga de informacion deberá ser menor a tres segundos.

**Requisitos del Hardware:**

* Procesador superior a 2.6Ghz de velocidad. Intel Pentium G4900 / AMD A6-3600
* Motherboard y cooler compatible al socket del procesador
* Minimo 4GB de memoria RAM
* Disco Solido de 128GB para un rápido inicio del sistema operativo
* Se recomienda una resolución de pantalla mínima de 1280 x 720 píxeles, ya que se necesitará suficiente espacio para visualizar los distintos elementos de la interfaz del usuario.
* Se necesitan dispositivos de entrada, como un mouse y un teclado, para interactuar con el software.
* Software: Windows 11.

**Requisitos no funcionales:**

RNFN01 - Realizar Login:

Se accede a esta función al abrir el programa o al realizar un Logout (RNFN02), siempre será necesario que no haya otro usuario logueado.

1. El sistema presentará una GUI para el Loguin (Inicio de sesión). En esta, el usuario ingresará el Nombre de Usuario y la contrseña
2. El sistema deberá indicar si algún campo está vacío
3. Si el usuario completa ambos campos, el sistema realizará la validación de las credenciales
4. Si los datos son verificados y son correctos, se ingresará al sistema principal.
5. Si los datos son verificados y no son correctos, se reducirá el contador de ‘Intentos fallidos’.

* Al completar tres intentos fallidos, se bloqueará al usuario
* Si se ingresan los datos correctos antes de antes de llegar a 3 intentos, se podrá ingresar al sistema
* Si se ingresan los datos correctos luego de realizar 3 intentos fallidos, se indicará al usuario que ha sido bloqueado y que debe comunicarse con un administrador

1. Se valida que permisos tiene el usuario en el sistema, el usuario tendrá acceso únicamente a las opciones que su rol permita.

RNFN02 - Realizar Logout:

Se accede a esta función dentro del sistema, cuando un usuario se encuentra logueado, dando la posibilidad de desloquearse,

1. Se contará con un botón “Logout” para cerrar la sesión
2. Una ventana emergente preguntará al usuario si está seguro de que quiere cerrar la sesión

* Si el usuario elige “Si”, se borrará todo lo que se encuentre en memoria
* Si el usuario elige “No”, se cerrará la ventana y continuará su sesión

1. Luego de cerrar sesión, se volverá a la ventana de Loguin

RNFN03 - Administrar perfiles:

Dentro del sistema, el usuario que tenga el rol de “Administrador”, tendrá la opción de poder gestionar los perfiles. Pudiendo generar nuevos perfiles, borrarlos, modificarlos, bloquearlos o desbloquearlos.

En la opción de modificar, se puede otorgar roles diferentes a los usuarios, lo que les permitirá realizar tareas diferentes dentro del sistema.

RNFN04 - Encriptar dato:

El sistema deberá presentar funciones de seguridad, una de ellas, será la encriptación de datos sensibles, como por ejemplo las contraseñas. Con el objetivo de que no se pueda acceder a ellas fácilmente, previniendo hackeos y perdidas de información.

RNFN05 - Cambiar idioma:

El sistema contendrá la funcionalidad de cambiar entre idiomas de forma dinámica, por defecto el programa estará en español, pero también incluirá las opción de inglés y portugués.

Con relación a los idiomas, los administradores tendrán la posibilidad de agregar o modificar los idiomas

RNFN06 - Guardar Backup:

Todos los Domingos (Día no laborable), el sistema deberá generar de manera automática un Backup de la base de datos, manteniendo la seguridad y la encriptación de los datos.

Otra forma de realizar un Backup, será manualmente por un usuario con el rol de administrador.

RNFN07 - Guardar en Bitácora:

El sistema deberá guardar en bitácora cada una de las acciones realizadas por un usuario o un sistema a lo largo del ciclo de vida del programa.

La bitácora incluirá: Nombre del usuario, descripción y fecha y hora.

RNFN08 - Generar Dígitos Verificadores:

Para asegurar la fiabilidad de los datos almacenados en la base de datos, se debe implementar un sistema de Verificación de Integridad de Datos que consiste en utilizar los dígitos de Verificación de Horizontalidad (DVH) y los dígitos de Verificación de Verticalidad (DVV). Cada tabla que contenga información crítica incluirá una columna para el DVH, y se generará una tabla separada para los DVV, que estará vinculada a la tabla original. Los dígitos se validarán al inicio del programa para garantizar la integridad de los datos en la base de datos

# G06 Diagramas de Clases y DER General

## G06.1 Diagrama de Clases general



G06.2 Diagrama de Clases general de la parte técnica y de servicios.



## G06.3 Diagrama de Clases general Por Capas



# G07. DER general tipo Martin

Diagrama

Descripción generada automáticamente

# N00. Procesos de negocio

# 

# N01. Especificación funcional por proceso de negocio

**Definicion de Roles**

* Recepcionista
  + Persona | Usa GUI
* Cliente
  + Persona | No usa GUI
* Banco
  + Sistema | Se relaciona con el Sistema

**N01-A Diagrama de Roles:**

Diagrama

Descripción generada automáticamente

**N01-B Diagrama de Secuencia de Roles**:

Diagrama

Descripción generada automáticamente

**Diagrama de Entrada, Comportamiento y Salida**



**Modelo de procesos:**

Diagrama

Descripción generada automáticamente

**Modelo conceptual:**



# N02. Especificaciones de Casos de Uso

**Modelo general de caso de uso:**



**Caso de Uso 20 – Generar Factura:** El recepcionista atiende el pedido de un cliente ingresando los productos en el carrito. El recepcionista ingresa el nombre y la cantidad del producto para que el sistema lo identifique, valide la disponibilidad según la cantidad ingresada y si todo es correcto se añada al carrito, luego de ingresar todos los productos, se procede a ingresar los datos complementarios (include “Ingresar datos complementarios”) y luego cobrar la venta (include “Cobrar venta”). Finalmente, se imprime la factura.

**Caso de uso 21 –Añadir datos complementarios:** Luego de terminar de cargar los productos, es necesario ingresar datos adicionales para continuar con la venta, estos mismos se incluirán en la factura de compra. Los datos son: número de mesa, nombre del mesero, comentarios adicionales, tipo de pedido (para llevar o para comer en el lugar).

**Caso de uso 22 – Cobrar venta:** Ya finalizados los pasos anteriores, se procede a completar la venta, el cliente tiene la opción de pagar en efectivo o con tarjeta. Si es con efectivo, se ingresa el monto y el sistema calcula el vuelto necesario, si es con tarjeta, se utiliza el posnet. De ser necesario, se procesa el pago con el banco.

|  |
| --- |
| **ID y Nombre**: CDU20 - **Generar Factura** |
| **Objetivo**: Representa el proceso iterativo de ingresar el pedido del cliente en el sistema, con el cual se genera una factura. |
|
| **Actor principal**: - Recepcionista |
| **Precondiciones**: El usuario debe estar logueado en el sistema y tener los permisos necesarios. |
|
| **Punto de extensión**: |
| **Disparador**: El recepcionista recibe un cliente y debe registrar los productos para la posterior venta. |
|
| **Postcondiciones**: Se ingresan los productos en el carrito y se continua la venta. |
| **Escenario principal**:   1. El recepcionista ingresa al menú “Realizar venta” 2. Se ingresa el Nombre y cantidad del producto 3. Se hace clic en el botón “Agregar” 4. El sistema valida el formato de los productos ingresados 5. El sistema valida que no falten datos 6. El sistema confirma la disponibilidad (stock) de los productos (producto) 7. Si es necesario ingresar más productos se retorna al punto 2 8. Cuando se termina de ingresar los productos al carrito se continúa completando los datos de la venta (RB)  Include “Ingresar datos de venta” 9. Al terminar el pedido se procede al cobro de este (RB)  Include “Cobrar venta” (RB) 10. El sistema actualiza el inventario automáticamente y guarda la base de datos (RB) |
|
| **Flujos alternativos**:   1. El formato de los productos no es valido 2. El sistema informa que faltan datos 3. El sistema confirma que no hay disponibilidad del producto. Volver a punto 2. 4. Al no estar disponible, no se agrega al carrito |

**Diagrama de secuencia:**



**Diagrama de Clases:**



Interfaz de usuario gráfica

Descripción generada automáticamente

|  |
| --- |
| **ID y Nombre**: CDU21 – **Ingresar datos de Venta** |
| **Objetivo**: Ingresar los datos restantes para completar la venta |
| **camviarActor principal**: - Recepcionista |
| **Precondiciones**: El usuario debe estar logueado en el sistema, tener los permisos necesarios y haber seleccionado los productos anteriormente. |
| **Punto de extensión**: - |
| **Disparador**: El recepcionista termina de llenar el carrito y debe ingresar datos de la venta |
| **Postcondiciones**: Se guardan los datos adicionales y se continua la venta |
| **Escenario principal**:   1. Luego de seleccionar los productos, se hace click en “Aceptar” y dispara la pantalla 2. Se muestran los campos a completar [número de mesa, nombre del mesero, comentarios adicionales, tipo de pedido (para llevar o para comer en el lugar)] 3. El usuario completa todos los campos y hace clic en “Finalizar Compra” 4. El sistema verifica que todos los campos estén completos y sean validos 5. Se mantiene la información. |
|
| **Flujos alternativos**:  3.1 – Los campos están vacíos y el sistema notifica al usuario  3.2 – Los campos no son válidos y el sistema notifica al usuario (Se deben ingresar caracteres alfabéticos) |

**Diagrama de secuencia:**



Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

|  |
| --- |
| **ID y Nombre**: CDU22 – **Cobrar venta** |
| **Objetivo**: Finalizar la venta, obteniendo el pago del cliente y emitiendo la factura correspondiente. |
| **Actor principal**: - Recepcionista |
| **Precondiciones**: El usuario debe estar logueado en el sistema, tener los permisos necesarios y haber Ingresado los datos de venta. |
| **Punto de extensión**: |
| **Disparador**: El recepcionista debe recibir el pago del cliente para poder entregar los productos y una factura de compra. |
| **Postcondiciones**: Se finaliza la compra y se entrega la factura. |
| **Escenario principal**:   1. Luego de ingresar los datos de venta, se hace click en “Aceptar” y se dispara la pantalla 2. El sistema muestra el listado de productos, información de la venta, monto total a pagar y permite seleccionar pago con efectivo o con tarjeta. 3. Al pagar con efectivo, se ingresa el monto otorgado por el cliente y se calcula el vuelto 4. El sistema valida que los campos sean correctos. 5. El recepcionista hace click en “Finalizar compra” (RB) 6. Se imprimen dos facturas, con la información de los productos y la información de venta. (RB) |
|
| **Flujos alternativos**:  3.1- Se paga con tarjeta y se utiliza el dispositivo.  3.2- Se confirma el pago y continua el punto 5 del flujo principal. (RB)  3.2.1- Se rechaza la tarjeta y se le comunica al usuario. Se retorna al punto 2. (RB) |

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



# RFN2 – gestión de compras:

N01.2- Definición de roles:

Proveedor (Persona – No actor directo – Sin GUI)

Encargado (Persona – Actor Directo – Con GUI)

Banco (Sistema – Se conecta al sistema – Sin GUI)

N01.2-A Diagrama de Roles



N01.2-A Diagrama de secuencia de roles



N01.2 Diagrama Entrada – Comportamiento – Salida



N01-C Diagrama de Procesos



N01.2-D Modelo Conceptual



**Diagrama de CASO DE USO:**



CDU81 – Analizar Existencias

|  |
| --- |
| **ID y Nombre**: CDU81 – Analizar Existencias |
| **Objetivo**: Detectar productos sin stock o con stock bajo. |
|
| **Actor principal**: - Encargado de Compras |
| **Precondiciones**: El usuario debe estar logueado y con los permisos necesarios. |
|
| **Punto de extensión**: CDU82 – Generar Cotización |
| **Disparador**: Se detectan producto/s a reponer stock. |
| **Postcondiciones**: Se genera un pedido de cotización al proveedor. |
| **Escenario principal**:   1. El usuario ingresa al área de gestión de compras, analizar existencia. Se despliega una grilla con todos los productos y su stock **(Nombre del producto, categoría, stock).** 2. Se ingresa un filtro y se actualizan los datos **(Filtro por nombre, categoría o cantidad de stock)** 3. El usuario realiza el análisis de existencia de los productos y selecciona aquellos deseados para realizar una cotización, luego da click en “Pedir cotización” 4. Se despliega una pantalla donde muestra los productos seleccionados, su stock actual y el proveedor al cual se enviará la cotización – Extend CDU82 – Generar cotización. |
| **Flujos alternativos**:  2.1- No se ingresa ningún filtro o el filtro ingresado no devuelve datos.  3.1- El usuario realiza el análisis de existencia de los productos y decide no realizar una cotización |

Diagrama de secuencia Analizar Existencias



Diagrama de clases analizar existencia



CDU82 – Generar cotización

|  |
| --- |
| **ID y Nombre**: CDU82 – Generar cotización |
| **Objetivo**: Generar una cotización de los productos deseados para enviar al proveedor. |
|
| **Actor principal**: - Encargado de Compras |
| **Precondiciones**: Se analizó la existencia de productos faltantes de stock y fueron seleccionados |
|
| **Punto de extensión**: - |
| **Disparador**: Se debe solicitar la cotización de productos a un determinado proveedor. |
| **Postcondiciones**: Se envia una cotización al proveedor. |
| **Escenario principal**:   1. Se despliega una pantalla donde muestra los productos seleccionados, su stock actual y el proveedor al cual se enviará la cotización 2. El usuario ingresa la cantidad que se desea solicitar de cada producto 3. El sistema genera un documento con la información del negocio, los productos solicitados y la cantidad. 4. Se genera un documento que se le otorga al usuario, conteniendo la información de la cotización |
| **Flujos alternativos**:  4.1- No se encuentra un proveedor asociado para reponer el producto, se notifica al usuario. |

Diagrama de secuencia Generar Cotizacion



Diagrama de clases Generar cotización



CDU83 – Generar Orden de Compra

|  |
| --- |
| **ID y Nombre**: CDU83 – Generar Orden de Compra |
| **Objetivo**: Generar la orden de compra para enviar al proveedor |
|
| **Actor principal**: - Encargado de Compras |
| **Precondiciones**: Se analizó la cotización previa y se selecciona generar orden de compra. Usuario logueado y con los permisos necesarios. |
|
| **Punto de extensión**: - |
| **Disparador**: Se desea generar la orden de compra. |
| **Postcondiciones**: Se envía una orden de compra al proveedor. |
| **Escenario principal**:   1. El usuario analiza la cotización y luego selecciona “Generar Orden de Compra” 2. El usuario ingresa los datos de la cotización dentro de la orden de compra (Datos del proveedor, datos del cliente, detalles de los productos) 3. El sistema verifica que la cotización sea válida (que el proveedor aun este registrado en el sistema, que los productos sean validos) 4. Se muestra un documento formal con la orden de compra. En donde aparece: La información de la empresa, la fecha, el número de orden de compra, información del proveedor, descripción de los productos, precios unitarios, cantidad, costo total, impuestos y descuentos, fecha de entrega estimada, información del envío e instrucciones especiales. 5. El usuario acepta la orden de compra y se genera un documento de Orden de Compra con la informacion proporcionada. (RB) 6. El sistema guarda un registro en la base de datos para una nueva compra con el estado de “Pendiente” en la base de datos |
| **Flujos alternativos**:  2.1- El sistema detecta un problema y lo notifica al usuario  4.1- El usuario rechaza la orden de compra |

Diagrama de Secuencia generar ordenes de compra



Diagrama de clases generar orden de Compra



CDU84 – Recibir Productos y Generar Factura

|  |
| --- |
| **ID y Nombre**: CDU84 – Recibir Productos y Generar Factura |
| **Objetivo**: Recibir los productos de una orden de compra |
|
| **Actor principal**: - Administrador |
| **Precondiciones**: Se debe haber generado una orden de compra, usuario logueado en el sistema con los permisos necesarios |
|
| **Punto de extensión**: - |
| **Disparador**: Se llega a la fecha de entrega de productos y se debe recibir la compra |
| **Postcondiciones**: Productos ingresados en el sistema. |
| **Escenario principal**:   1. El usuario accede al sistema y selecciona la opción Gestionar compras -> "Recibir Productos". 2. El sistema despliega una lista de las órdenes de compra pendientes de recepción, mostrando los detalles de la orden de compra como fecha, proveedor, productos solicitados y cantidades. 3. El usuario selecciona la orden de compra que corresponde a los productos recibidos. 4. El sistema carga automáticamente los detalles de la orden de compra, incluyendo los productos solicitados y sus cantidades. 5. El usuario verifica visualmente los productos físicos recibidos y los compara con la información de la orden de compra. 6. Los productos coinciden con lo solicitado y el usuario procede a confirmar la recepción. 7. El sistema actualiza la compra como finalizada. El sistema actualiza productos y stock (RB) 8. Se genera una factura con los datos de la orden de compra. La factura se crea automáticamente utilizando los datos de la orden de compra y se asocia con el proveedor y los productos recibidos. |
| **Flujos alternativos**:  5.1- si el usuario identifica alguna discrepancia entre los productos físicos y la orden de compra, puede registrar estas discrepancias en el sistema y cambiar la cantidad de productos recibida. |

Diagrama de secuencia de Recibir Productos y Generar Factura



Diagrama de Clases Recibir Productos y Generar Factura



# T00. Documentos de aspectos técnicos que provee el sistema de información.

## T01. Arquitectura Base



**BE (Entidades):**

Descripción: La capa de Entidades (BE - Business Entities) contiene las clases que representan objetos de negocio puros y simples. Estas clases suelen reflejar directamente las estructuras de datos en la base de datos o los objetos que se manejan en la aplicación.

Responsabilidades: Definir la estructura de los objetos de negocio sin lógica de negocio. Pueden contener propiedades y métodos relacionados con la validación de datos o la transformación de datos.

**Vista**:

Descripción: La capa de Vista es la interfaz de usuario de la aplicación. Aquí se encuentran las páginas web, formularios de Windows, componentes gráficos y otros elementos que permiten a los usuarios interactuar con la aplicación.

Responsabilidades: Presentar datos al usuario, capturar la entrada del usuario y mostrar resultados. Generalmente, no debe contener lógica de negocio.

**BLL (Negocio):**

Descripción: La capa de Negocio (BLL - Business Logic Layer) es donde reside la lógica de negocio de la aplicación. Contiene clases y componentes que procesan datos, aplican reglas de negocio y coordinan las acciones entre las Entidades y la capa de acceso a datos (DAO).

Responsabilidades: Validar datos, implementar reglas de negocio, gestionar transacciones y coordinar la interacción entre las capas.

**DAO (Acceso a Datos):**

Descripción: La capa de Acceso a Datos (DAO - Data Access Object) se encarga de interactuar con la base de datos o cualquier otro sistema de almacenamiento de datos. Aquí se escriben consultas, se establecen conexiones y se realizan operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en la base de datos.

**Responsabilidades**: Acceder y manipular datos en el almacenamiento persistente, manejar conexiones a la base de datos, mapear datos entre objetos de entidad y estructuras de almacenamiento.

Servicios:

**Descripción**: La capa de Servicios proporciona funcionalidades específicas de la aplicación que pueden ser compartidas entre diferentes partes de la aplicación o incluso entre aplicaciones diferentes. Esto podría incluir servicios web, servicios de autenticación, servicios de notificación, etc.

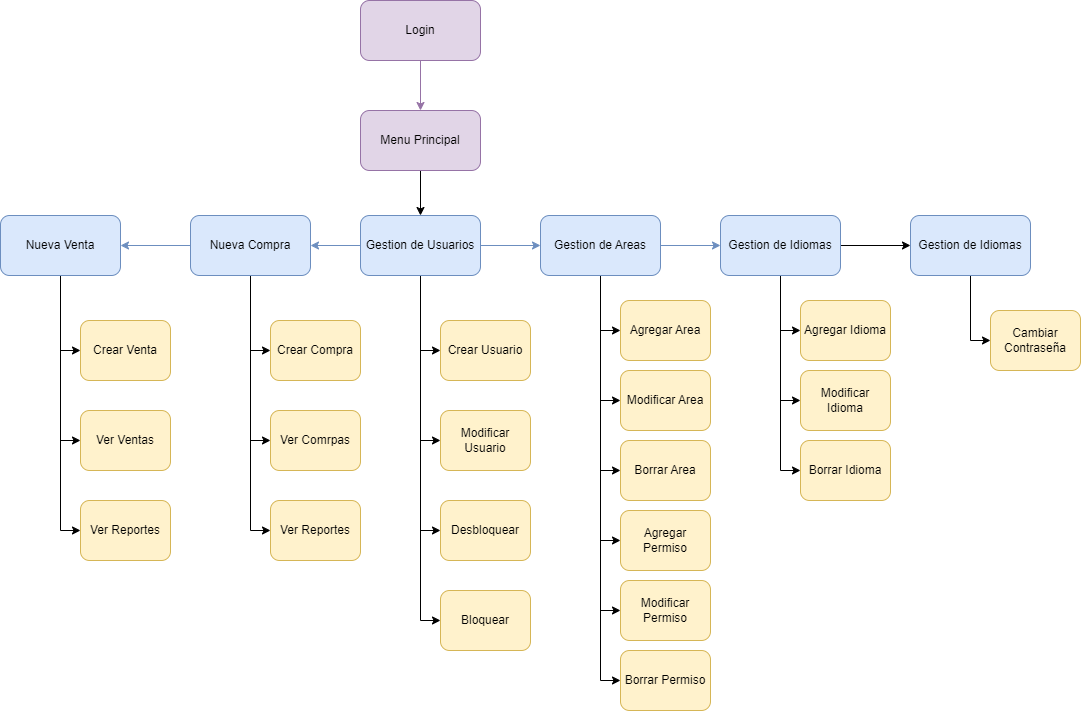
Responsabilidades: Proporcionar funcionalidades reutilizables o servicios aislados que pueden ser invocados por otras partes de la aplicación. Pueden encapsular lógica compleja y ofrecer una interfaz estandarizada para su uso.





# Mapa de Navegación





# T02.1 Gestión de usuarios. Crear

**Descomposición funcional Crear Usuario**

1. Se ingresa a "gestión de usuarios" -> "Crear usuario" ubicado en el lateral del menú
2. El sistema solicita que ingrese El email, nombre, apellido y Perfil
3. El sistema comprobará que no exista ningún usuario previamente registrado en la base de datos con el mismo email.
4. Una vez que todos los campos estén ingresados y sean validados, el sistema creará un nuevo usuario utilizando la información proporcionada por el administrador.



**Especificación funcional CDU03 – Crear Usuario:**

|  |
| --- |
| **ID y Nombre: CDU03** – Crear Usuario |
| **Objetivo:** Permitir al administrador del sistema registrar un nuevo usuario con los datos pertinentes, asignarle un área específica y almacenar la información en la base de datos. |
|  |
|  |
| **Actor principal:** Usuario Administrador |  |
| **Precondiciones** El usuario debe estar autenticado en el sistema como usuario con privilegios para crear usuarios. |  |
| **Punto de extensión: -** |  |
| **Disparador:** El Administrador necesita un nuevo perfil para asignarle a un empleado, y así ingresar al sistema |  |
| **Postcondiciones:** Se crea un nuevo registro de usuario, con la información proporcionada por el administrador, como nombre de usuario y contraseña.  El usuario recién creado se asocia con un área específica, determinando los permisos y acciones que puede realizar en el sistema |  |
|  |
| **Escenario principal:**   1. Se ingresa a "gestión de usuarios" -> "Crear usuario" ubicado en el lateral del menú 2. El sistema despliega una lista de usuarios existentes y muestra opciones, como crear un nuevo usuario, modificar un usuario, bloquear o desbloquear. 3. El sistema solicita que ingrese los campos necesarios para crear el nuevo usuario (Nombre, Apellido, Email, Perfil) 4. El sistema valida que todos los campos estén completos 5. El administrador selecciona la opción "Crear usuario" para continuar con el proceso. 6. Se confirma que los datos ingresados son correctos y válidos (No existe otro usuario con ese Mail, los caracteres y sintaxis son válidas). 7. El sistema genera una contraseña automáticamente (nombre + apellido), la encripta y se guarda en la base de datos (RB) |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **Flujos alternativos:**  4.1- No están los campos completos y el sistema notifica  6.1- Los datos correctos no son correctos o válidos y se notifica. |  |

**Diagrama de secuencia crear usuario**



**Diagrama de clases crear usuario**



**DER Crear usuario**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

**GUI Crear usuario**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

# T02.2 Gestión de Log In

**Descomposición Funcional Login:**

1. Para acceder al sistema, El usuario ingresa un nombre de usuario y una contraseña.
2. El sistema verifica que el usuario y contraseña ingresados sean correctos y se encuentren en la base de datos. Si los datos son incorrectos, se mostrará un mensaje de error y se dará la opción de volver a ingresar las credenciales.
3. Si un usuario ingresa información incorrecta tres veces seguidas, su cuenta quedará bloqueada.
4. Si un usuario está bloqueado, no podrá iniciar sesión incluso si ingresa la información correcta.
5. Si un usuario ingresa correctamente su información de inicio de sesión (Y no está bloqueado), se inicia sesión y el contador de intentos se reiniciará a cero.
6. Se mostrarán solo las funcionalidades a las que tengan acceso.

**Diagrama de caso de uso Login:**

Diagrama

Descripción generada automáticamente

**Especificación funcional Login:**

|  |
| --- |
| **ID y Nombre: CDU01** – Iniciar Sesión |
| **Objetivo:** El usuario necesitaIngresar al sistema para operar el negocio, |
| **Actor principal:** Empleado |  |
| **Precondiciones:** Usuario desbloqueado.No debe haber una sesión ya iniciada |  |
| **Punto de extensión: CDU02 – Cambiar Idioma** |  |
| **Disparador:** El usuario necesita realizar una venta o una compra en el sistema. |  |
| **Postcondiciones:** El usuario se ingresa al sistema y tiene acceso a las actividades que le corresponden. |  |
|  |
| **Escenario principal:**   1. Se muestra la pantalla de Login, donde el usuario ingresa sus datos (Email y Contraseña). 2. Al hacer clic en ingresar, Se comprueba que el usuario no esté ya logueado 3. Se encripta la contraseña y se validan las credenciales con las de la base de datos. 4. Los datos del usuario son correctos, Se comprueba que no esté bloqueado y que el contador de intentos sea menor a 3. 5. Se restablece el contador de intentos 6. Se inicia sesión en el sistema y se muestran los accesos correspondientes según su perfil (RB) 7. El sistema realiza la gestión del digito verificador, y si es correcto, se concede el acceso al sistema |  |
|  |
|  |
|  |
|  |
|  |
|  |
| **Flujos alternativos:**   * 1. El usuario decide cambiar de idioma – Extends CDU02 – Cambiar idioma   3.1- Hay un error de conexión con la base de datos. Se informa por pantalla que hubo un error  3.2- Hay un error en las credenciales ingresadas. Se informa al usuario “Volver a ingresar credenciales” y se aumenta el contador de intentos incorrectos. Se vacían los textbox y se retorna al punto 2  3.3- Si se introduce un usuario válido pero se equivoca en la contraseña tres veces, se bloqueará y se informará al usuario. Luego, se vaciarán los campos de texto y se volverá al paso dos.  3.4- En caso de que se ingrese un usuario válido con la contraseña correcta pero dicho usuario esté bloqueado, se le notificará al usuario que debe ponerse en contacto con el administrador y se regresará al punto dos.  3.5- Si se ingresan las credenciales correctas y el usuario no está bloqueado, pero ya hay una sesión abierta, se le comunicará al usuario y se regresará al punto dos.  7.1- Si se detecta una discrepancia en alguna tabla que es administrada por el dígito verificador, el sistema denegará el acceso a todos los usuarios y únicamente permitirá que el administrador ingrese para resolver el problema. |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Diagrama de secuencia Login:**



**Diagrama de Actividad Login:**

Gráfico, Diagrama

Descripción generada automáticamente

**Diagrama de Clases Login:**

****

**Diagrama DER Login:**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media**

**GUI Login:**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

# T02.3 Desbloquear Usuario

**Descomposición funcional Desbloquear usuario**

1. El administrador ingresa en “gestión de usuarios” ubicado en el lateral del menú principal
2. El sistema mostrará una lista de los usuarios ya ingresados en el sistema
3. El administrador selecciona el usuario
4. El administrador seleccionará el botón "Desbloquear usuario".
5. El sistema verificará si el usuario está bloqueado antes de confirmar la acción
6. Al confirmar la acción el usuario dejará de estar bloqueado, por lo que puede ingresar al sistema.

**Diagrama de caso de uso Desbloquear Usuario**



**Especificación funcional CDU05 –Desbloquear Usuario:**

|  |
| --- |
| **ID y Nombre: CDU05** – Desbloquear Usuario |
| **Objetivo:** Permitir al administrador del sistema Desbloquear usuario que se encuentra bloqueado del sistema, permitiendo que vuelva a ingresar. |
|  |
|  |
| **Actor principal:** Usuario Administrador |  |
| **Precondiciones** El usuario debe estar autenticado en el sistema como usuario con privilegios para crear usuarios. |  |
| **Punto de extensión: -** |  |
| **Disparador:** El Administrador necesita desbloquear al usuario para que pueda ingresar al sistema |  |
| **Postcondiciones:** Se desbloquea el usuario en el sistema, permitiendo el acceso del mismo. |  |
|  |
| **Escenario principal:**   1. El sistema muestra una lista de usuarios existentes y presenta opciones para administrarlos. 2. El administrador selecciona un usuario 3. El administrador hace clic en desbloquear 4. El sistema muestra una confirmación para desbloquear y se selecciona confirmar. 5. El usuario se encontraba bloqueado, el sistema confirma la acción de desbloqueo y realiza los cambios necesarios en la configuración del usuario en la BBDD. (RB) 6. El sistema muestra un mensaje que indica que el usuario ha sido desbloqueado exitosamente. |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **Flujos alternativos:**  4.1- El sistema muestra una confirmación para desbloquear y se selecciona cancelar.   * 1. El usuario no estaba bloqueado, el sistema avisa que no se pudo realizar ninguna acción. (RB) |  |





Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

# T02.4 Cambiar Contraseña

**Descomposición Funcional Cambiar Contraseña**

1. Para cambiar la contraseña el usuario deberá hacer clic en el botón “Perfil” y en “Cambiar contraseña”
2. Completa los campos “Ingresar contraseña actual”, “nueva contraseña”, y “Confirmar Contraseña”
3. El usuario completará los campos y aceptará.
4. El sistema comprueba que la contraseña actual sea la correcta, y que la nueva contraseña y confirmar contraseña coincidan.
5. Si pasa todas las medidas de seguridad, se muestra un mensaje de confirmación al usuario
6. Se cambian los datos del usuario y se guarda en la base de datos



**Especificación funcional CDU06 –Cambiar Contraseña:**

|  |
| --- |
| **ID y Nombre: CDU06** – Cambiar Contraseña |
| **Objetivo:** Permitir a los usuarios cambiar su contraseña para ingresar al sistema |
|  |
|  |
| **Actor principal:** Usuario |  |
| **Precondiciones** El usuario debe estar autenticado en el sistema como usuario. |  |
| **Punto de extensión: -** |  |
| **Disparador:** El usuario requiere cambiar la contraseña |  |
| **Postcondiciones:** El usuario modifica su contraseña y el usuario ahora ingresara al sistema con su nueva contraseña |  |
|  |
| **Escenario principal:**   1. El usuario selecciona “Cambiar Contraseña” 2. El usuario completa los campos y acepta el cambio de contaseña (Contraseña actual, nueva contraseña, repetir nueva contraseña) 3. El sistema encripta la contraseña actual 4. El sistema compara la contraseña actual ingresada con la base de datos y es correcta 5. El sistema compara nueva contraseña y repetir contraseña y coinciden 6. El sistema encripta la nueva contraseña y la guarda en la base de datos (RB). 7. Los campos son válidos y se muestra un mensaje de confirmación. |  |
|  |
|  |
|  |
| **Flujos alternativos:**  2.1- El usuario no completa todos los campos y lo notifica el sistema.  4.1.1- El sistema compara la contraseña actual ingresada con la base de datos y es incorrecta  4.1.2-Se notifica al usuario  5.1.1- El sistema compara nueva contraseña y repetir contraseña y no coinciden  5.1.2- Se notifica al usuario |  |





Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

# T02.5 Gestión de Logout

**Descomposición funcional del Logout**

1. Para cerrar sesión, el usuario hará clic en el botón “Cerrar sesión” ubicado en el panel lateral
2. El sistema muestra una ventana la cual contiene un botón de cancelar y otro de aceptar
3. El usuario confirma cerrar sesión.
4. Se muestra la pantalla de Login



**Especificación funcional CDU07 – Realizar Logout:**

|  |
| --- |
| **ID y Nombre: CDU03** – Realizar Logout |
| **Objetivo:** Cerrar sesión en el sistema |
| **Actor principal:** Usuario |  |
| **Precondiciones** El usuario debe estar autenticado en el sistema. |  |
| **Punto de extensión: -** |  |
| **Disparador:** El usuario debe salir del sistema |  |
| **Postcondiciones:** Se cierra sesión y se borra caché |  |
| **Escenario principal:**   1. El sistema muestra la ventana de confirmación para cerrar sesión. 2. El usuario presiona el botón cerrar sesión 3. Se muestra la GUI de cerrar sesión la cual tiene el botón de cancelar y aceptar 4. Al presionar aceptar, se cierra sesión y se borra cache 5. Se eliminan todos los datos de la sesión 6. Se muestra la pantalla de Login |  |
|  |
|  |
|  |
| **Flujos alternativos:**  3.1. Se presiona cancelar y se continua con la sesión |  |





Diagrama De Clases



Tabla

Descripción generada automáticamente

# T03. Gestión de Encriptado

## T03.1 Encriptacion Irreversible

En el sistema de gestión para Dialect Café, hemos implementado una medida de seguridad importante conocida como encriptación irreversible. Esta técnica nos permite proteger las contraseñas de los usuarios de manera segura, evitando que estén almacenadas en texto plano y expuestas a posibles amenazas.

Lo que hacemos es aplicar un proceso de encriptación llamado 'hashing' a las contraseñas ingresadas por los usuarios. Utilizamos un algoritmo específico llamado bcrypt, que es conocido por ser seguro y resistente a ataques de fuerza bruta. Este algoritmo genera un valor irreversible de longitud fija a partir de la contraseña original.

Cuando un usuario crea una cuenta o cambia su contraseña, la contraseña ingresada se somete al proceso de encriptación mediante bcrypt. El resultado obtenido, el 'hash', se almacena en nuestra base de datos. Es importante destacar que este hash es único para cada contraseña y no se puede revertir fácilmente al valor original.

Cuando un usuario intenta iniciar sesión, tomamos la contraseña ingresada y la encriptamos nuevamente utilizando bcrypt. Luego comparamos este nuevo hash generado con el valor almacenado en la base de datos. Si ambos hashes coinciden, el sistema considera que la contraseña es válida y permite el acceso al usuario.

Con este enfoque, garantizamos que las contraseñas de nuestros usuarios estén protegidas de manera segura. Incluso en el caso de que alguien acceda a nuestra base de datos, sería extremadamente difícil revertir el hash y obtener la contraseña original. Esto brinda tranquilidad tanto a los usuarios como a nosotros como administradores del sistema.

Diagrama, Texto, Pizarra

Descripción generada automáticamente

## T03.2 Encriptación Reversible

En el sistema de gestión para Dialect Café, hemos implementado una medida de seguridad adicional llamada encriptación reversible. Esta técnica nos permite proteger datos sensibles, como nombres de usuario, información de productos o información de proveedores, de manera segura y confiable.

Para lograrlo, utilizamos un algoritmo de encriptación reversible conocido como Advanced Encryption Standard. AES es ampliamente aceptado y proporciona una fuerte seguridad para proteger nuestros datos. Lo mejor de todo es que nos permite recuperar los datos originales cuando sea necesario.

Con la implementación de AES, aplicamos la encriptación reversible a los datos sensibles. Esto significa que, al guardarlos en nuestra base de datos, se transforman en una forma encriptada y segura que solo puede ser revertida utilizando la clave correspondiente.

Cuando necesitamos acceder a estos datos, como para mostrar la información de un proveedor o consultar los detalles de un producto, desencriptamos selectivamente los datos utilizando la clave adecuada. Esto nos permite acceder a la información original de manera segura y protegerla de accesos a terceros.

Diagrama

Descripción generada automáticamente

# T04. Gestión de Perfiles de Usuario

Crear Perfil

**Descomposición Funcional:**

1. El usuario administrador ingresa a “gestión de Perfiles”
2. Se muestra una grilla con los perfiles ya creados, una grilla con los permisos compuestos y una grilla con permisos simples. Hay botones para ABM de perfiles y permisos.
3. Al seleccionar un permiso compuesto, se despliegan los permisos que lo componen.
4. El administrador selecciona “Agregar Perfil”, donde ingresa el nombre del perfil y selecciona un permiso compuesto
5. Al guardar, se crea el perfil, el cual se puede asignar a los usuarios en la sección de “gestión de usuarios”

**Diagrama de caso de uso**



**Especificación funcional CDU30 – Crear Perfil**

|  |
| --- |
| **ID y Nombre: CDU30 – Crear Perfil** |
| **Objetivo:** Gestionar los perfiles del sistema con los que los usuarios ingresan la sesión. |
| **Actor principal:** Usuario Administrador |  |
| **Precondiciones** El usuario debe estar autenticado en el sistema. Con los permisos necesarios |  |
| **Punto de extensión: -** |  |
| **Disparador:** El administrador debe crear un nuevo perfil. |  |
| **Postcondiciones:** Se crea un nuevo perfil |  |
| **Escenario principal:**   1. El usuario con permisos de administrador ingersa a la pestaña de “gestión de perfiles” 2. En el sistema se observa una tabla con Perfiles, una con Permisos Compuestos y otra con Permisos Simples, junto con los botones ABM. 3. El usuario selecciona un PC y hace clic en agregar perfil. 4. Se solicita un nombre para el nuevo perfil 5. El usuario ingresa un nombre 6. Se crea un nuevo perfil, con un id, el nombre ingesado por el usuario y su código de permiso compuesto 7. Se informa al usuario la acción exitosa 8. Se guarda en base de datos y se actualizan las vistas (RB) |  |
|  |
|  |
|  |
| **Flujos alternativos:**  4.1- No se completan todos los compas y el sistema lo notifica. |  |

**Diagrama de secuencia Crear Perfil**



**Diagrama de clases**



**Diagrama entidad relación**

Diagrama

Descripción generada automáticamente

Rectángulo

Descripción generada automáticamente

# ASIGNAR PERFIL A USUARIO

**Descomposición funcional de asignar perfil:**

1. El administrador ingresa en “Gestion de usuarios”
2. El usuario selecciona modificar un perfil
3. El administrador selecciona el perfil desde el combo box de “Perfiles”
4. El administrador guarda los cambios y se asigna el perfil

**Diagrama de caso de uso Asignar Perfil:**



**Especificación funcional CDU32 – Asignar Perfil**

|  |
| --- |
| **ID y Nombre: CDU30 – Asignar Perfil** |
| **Objetivo:** Asignar perfiles a los usuarios |
| **Actor principal:** Usuario Administrador |  |
| **Precondiciones** El usuario debe estar autenticado en el sistema. Con los permisos necesarios |  |
| **Punto de extensión: -** |  |
| **Disparador:** El administrador debe asignar un perfil a un usuario. |  |
| **Postcondiciones:** Se modifica el permiso de un usuario |  |
| **Escenario principal:**   1. El administrador ingresa a “Gestion de Usuarios” 🡪 “Modificar usuario” 2. Se muestra una grilla con todos los usuarios, el administrador selecciona el usuario a modificar 3. El administrador selecciona un perfil del comboBox 4. El sistema valida que los campos sean correctos 5. Se ingresa en la base de datos el usuario con su perfil correspondiente y se informa la operación exitosa (RB) |  |
|  |
|  |
|  |
| **Flujos alternativos:**  4.1 El sistema detecta datos inválidos y se comunica al usuario |  |

**Diagrama de Secuencia Asignar Perfil:**



**Diagrama de clases asignar perfil:**



**Diagrama entidad relación:**

Diagrama

Descripción generada automáticamente

# T05. Gestión de Múltiples Idiomas

**Descomposición Funcional**

1. El administrador ingresa a la gestión de Idiomas
2. Se visualiza La lista de los idiomas, una grilla con los datos de las traducciones (TraduccionID, nombre idioma, TagID, Tag Nombre, Descripcion y Texto), las opciones de Agregar Idioma, Eliminar Idioma, Editar Traduccion, Agregar Tag, Eliminar Tag.
3. El administrador selecciona “Agregar Idioma”, se despliega un input e ingresa el nombre del idioma.
4. En la grilla se visualiza el nuevo idioma y todas las traducciones con el texto “-----“.

Diagrama de Casos de uso Crear Idioma



Especificación de caso de uso Crear Idioma

|  |
| --- |
| **ID Y NOMBRE:** CDU81 – Crear Idioma |
| **OBJETIVO:** El administrador crea un nuevo idioma |
| **ACTOR PRINCIPAL:** Administrador |
| **ACTORES SECUNDARIOS: --** |
| **PRECONDICIONES:**  El empleado debe tener los permisos necesarios para crear un nuevo idioma |
| **PUNTOS DE EXTENSIÓN: --** |
| **DISPARADOR:** Se desea ingresar un nuevo idioma en el sistema |
| **ESCENARIO PRINCIPAL:**   1. El administrador ingresa en la gestión de idiomas 2. Se visualiza La lista de los idiomas, una grilla con los datos de las traducciones (TraduccionID, nombre idioma, TagID, Tag Nombre, Descripcion y Texto), las opciones de Agregar Idioma, Eliminar Idioma, Editar Traduccion, Agregar Tag, Eliminar Tag 3. El administrador selecciona “Agregar Idioma”, se despliega un input e ingresa el nombre del idioma. 4. El sistema valida que no exista el idioma anteriormente. 5. En la base de datos se ingresa el nuevo idioma (RB) 6. Se generan automáticamente las traducciones, se relacionan todos los tags con el nuevo idioma, con el texto traducido en “-----“ 7. En la grilla se visualiza el nuevo idioma y todas las traducciones con el texto “-----“. 8. El administrador ingresa los textos para cada traducción – Extend – CDU82 |
| **FLUJOS ALTERNATIVOS:**  5.1 Ya existía un Idioma con el mismo nombre, el sistema solicita un reingreso, retorna al punto 4 |
| **POSTCONDICIONES:** El usuario cargo el idioma con exito |

**Diagrama de secuencia Crear Idioma**

****

**Diagrama de Clases Crear Idioma**

****

**Der Crear Idioma**

****

**Especificación de caso de uso Editar Idioma**

|  |
| --- |
| **ID Y NOMBRE:** CU 82 – Editar Idioma |
| **OBJETIVO:** El empleado edita un idioma |
| **ACTOR PRINCIPAL:** Empleado Administrador |
| **ACTORES SECUNDARIOS: --** |
| **PRECONDICIONES:**  El empleado debe tener los permisos necesarios para modificar un idioma |
| **PUNTOS DE EXTENSIÓN:** CU 81 – crear Idioma |
| **DISPARADOR:** El empleado selecciona una traducción de la grilla, la edita y presiona el botón guardar traducción |
| **ESCENARIO PRINCIPAL:**   1. El administrador selecciona del combo box un idioma a editar 2. El sistema carga en el datagrid todos tags del sistema, su descripción y su traducción 3. El empleado selecciona una traducción y selecciona editar traducción 4. Escribe la traducción y presiona el botón guardar 5. El sistema actualiza en la base de datos, en la tabla IdiomaControl la traducción del botón que se modificó [RB] |
| **FLUJOS ALTERNATIVOS:** |
| **POSTCONDICIONES:** El empleado modifico el idioma con éxito. |

**Diagrama de secuencia Editar Idioma**

****

**Diagrama de clases Editar Idioma**

****

**Der Editar Idioma**

****

# T06A Gestion de Bitacora de Evento

• Introducción:

En esta bitácora queda un registro de las operaciones que los usuarios realizan utilizando el sistema, lo que permite tener un historial de las actividades que cada usuario realiza en la aplicación.

De esta manera, la gestión de la bitácora cuenta con dos dimensiones:

* Dimensión Interna:

Donde se guardan los datos cuando los usuarios interactúan con el sistema

* Dimensión Externa:

Un usuario puede consultar los datos de la bitácora.

**DIMENSION INTERNA BITACORA EVENTOS:**

1. El usuario realiza alguna acción sobre el sistema
2. El sistema captura los datos de la operación realizada, la fecha y hora, el usuario, el módulo donde se realizó la acción, y la criticidad de la operación.
3. Se genera un registro en la base de datos.

**DIMENSION EXTERNA BITACORA EVENTOS:**

1. El usuario administrador ingresa dentro de bitácora -> bitácora eventos
2. El sistema muestra los datos en la grilla (operación realizada, la fecha y hora, el usuario, módulo y criticidad)
3. El usuario aplica filtro, la grilla se filtra según lo ingresado.

Registro de Eventos – Dimension Interna

• Funcionamiento:

La bitácora de eventos registra las acciones realizadas por los usuarios en el sistema. Las acciones que deben ser registradas en la bitácora se encuentran definidas en las especificaciones de casos de uso como RB (Reportar a Bitacora).

Ejemplo:

Cuando un usuario inicia sesión en el sistema -> RB (Se registra esta acción como "Login" en la bitácora)

• Captura de Datos:

La bitácora de eventos captura los siguientes datos en el momento en que un usuario realiza una acción:

Mail del Usuario, fecha, acción realizada, Módulo donde se realizó la acción y Criticidad.

**Mail del Usuario:** Se obtiene a través del sessionManager y corresponde al correo del usuario registrado en el sistema.

**Fecha:** Se registra la fecha y hora exactas en las que se realizó la acción.

**Acción:** Se registra la acción específica realizada por el usuario en el sistema. Por ejemplo, "Crear nueva Venta", "Login" o "Logout".

**Módulo:** Se identifica la parte del sistema en la que se llevó a cabo la acción. Por ejemplo, "Carrito de Ventas".

**Criticidad:** Se asigna un nivel de impacto en el sistema, evaluado en una escala del 1 al 3. El nivel 3 representa un impacto bajo, mientras que el nivel 1 indica un impacto alto en el sistema.

Ejemplos de criticidad:

* Nivel 1: Se refiere a acciones de máxima criticidad, como la modificación de registros en la base de datos. Ejemplos incluyen la creación de una nueva venta o la creación de un nuevo usuario.
* Nivel 2: Indica una criticidad media, donde no se registran cambios en la base de datos, pero se ingresan datos sensibles en el sistema. Ejemplos son la introducción de datos de venta o la selección de productos en el carrito.
* Nivel 3: Representa una criticidad baja y abarca acciones comunes que no afectan los datos ni el funcionamiento del negocio. Ejemplos incluyen el inicio de sesión (Login) y la salida del sistema (Logout).

• Base de Datos:

Tabla

Descripción generada automáticamente

**Diagrama de secuencia reportar bitácora**



**Diagrama de clases reportar bitácora**

****

**DER Reportar bitácora**

Tabla

Descripción generada automáticamente

## Auditar bitácora de Eventos – Dimensión Externa

**Descomposición Funcional**

1. El administrador ingresa en el menú Bitacoras -> Bitacora de Eventos
2. Se muestran una grilla donde se reportan las acciones capturadas por la bitácora y una sección para filtrar la grilla
3. El usuario selecciona el campo por el cual quiere filtrar e ingresa datos en el campo de búsqueda
4. El usuario selecciona buscar y se filtra la grilla

**Diagrama de caso de uso**



**Especificación funcional CDU41 – bitácora de Eventos** – Dimensión Externa

|  |
| --- |
| **ID y Nombre: CDU41 – bitácora de Eventos** – Dimensión Externa |
| **Objetivo:** Auditar las actividades de los usuarios en el sistema |
| **Actor principal:** Usuario Administrador |  |
| **Precondiciones**: |  |
| **Punto de extensión: -** |  |
| **Disparador:** El usuario realiza una acción en el sistema. |  |
| **Postcondiciones:** Se registra la bitácora. |  |
| **Escenario principal:**   1. El usuario con rol de administrador ingresa en bitácora -> bitácora Eventos 2. El sistema muestra los datos de la bitácora de eventos (la operación realizada, la fecha y hora y el usuario) en una grilla. 3. El usuario selecciona algún filtro: (la operación realizada, la fecha y hora y el usuario), ingresa los datos en un campo de texto y presiona Buscar 4. El sistema filtra la grilla con los datos de la bitácora. |  |
|  |
| **Flujos alternativos:**  3.1- El usuario no ingresa ningún filtro, el sistema cancela la operación.  4.1- El filtro ingresado no contiene datos. Se notifica al usuario. |  |

**Diagrama de Secuencia:**



**Diagrama de Clases:**

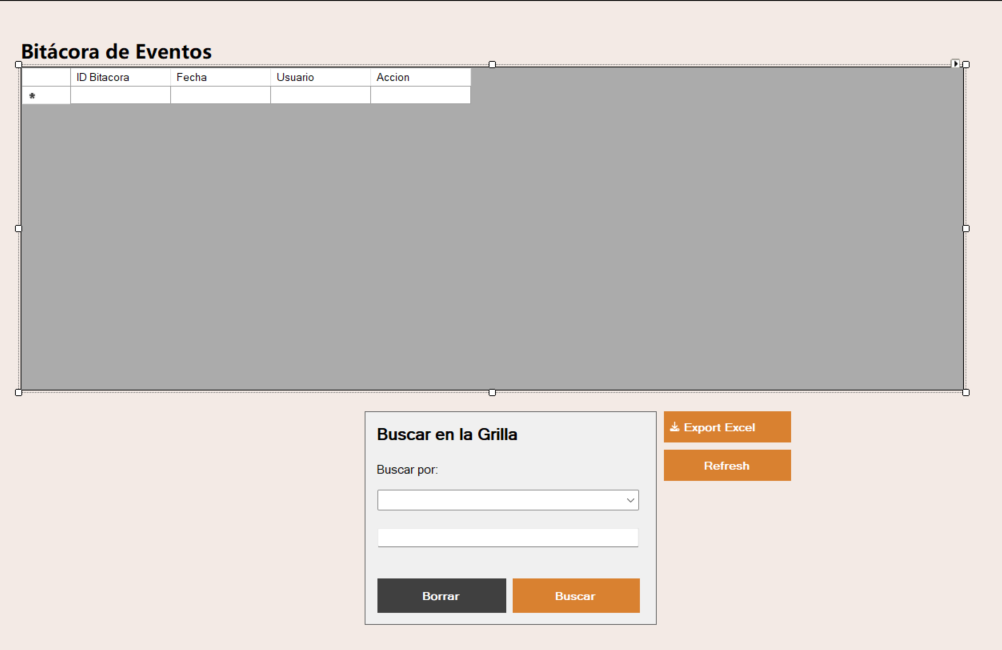
****

**Diagrama Entidad relación:**

Tabla

Descripción generada automáticamente

**GUI Auditar Bitacora de Eventos:**



# T06b. Bitacora de cambios

• Introducción:

El control de cambios busca tener un historial mucho más específico sobre las acciones de los usuarios en el sistema. Esta bitácora guarda un registro de las acciones realizadas por los usuarios en determinadas tablas del sistema, sin necesidad de intervención del usuario.

La bitácora de cambios se utilizará sobre la tabla Usuarios, de forma tal que, cada vez que se realice una acción de inserción, modificación o borrado sobre la tabla, se genera un registro en la tabla de bitácora, en la cual se detalla: El usuario que realizó la acción, tipo de acción, dato previo a la acción, dato posterior a la acción, fecha y hora, comando SQL realizado y el comando SQL para deshacer el cambio.

Existen dos dimensiones para el control de cambios:

* **Dimensión Interna:** Es el proceso en donde se registran las acciones de los usuarios en la base de datos.
* **Dimensión Externa:** Corresponde al formulario con el cual el usuario puede visualizar la bitácora de cambios y, además, puede cambiar a algún estado previo de los registros.

**Dimensión Interna Bitácora Cambios:**

1. El usuario realiza alguna acción sobre el sistema
2. El sistema captura los datos de la operación (fecha y hora, Mail del usuario, dato anterior, dato actual, tabla donde se realizó el cambio)
3. El sistema guarda el nuevo registro en la base de datos.

**Dimensión Externa Bitácora Cambios:**

1. El sistema muestra una pantalla donde se visualizan los datos de la bitácora de cambios en una grilla.
2. En esta pantalla, se puede filtrar la grilla por cualquiera de sus campos.
3. Si el filtro ingresado no contiene datos. Se notifica al usuario.

• Funcionamiento:

La bitácora de cambios, en su funcionamiento, registra las acciones del usuario en la tabla de usuarios. Esto incluye inserciones, modificaciones y eliminaciones de usuarios en el sistema.

Ejemplo:

Cuando un administrador introduce un Nuevo Usuario en el sistema, esta acción se registra en la bitácora de cambios.

• Captura de Datos:

La bitácora de cambios captura los siguientes datos en el momento en que un usuario realiza una acción:

Usuario, acción, dato Previo, dato Posterior, fecha y hora, SQL ejecutado, SQL de reversa.

**Mail del Usuario:** Contiene el mail del usuario que realizo la acción. Se obtiene a través de las variables de la sesión de la base de datos.

**Acción:** Se registra la acción específica realizada por el usuario en el sistema. Pueden ser: ‘INSERT’, ‘UPDATE’ o ‘DELETE’.

**Dato Previo:** En caso de que haya un UPDATE o DELETE, Se capturan los datos previos a la modificación. En caso contrario (Se realizó un INSERT) no habrá datos previos, por lo que se ingresará “Sin datos previos”

**Dato Posterior:** En caso de que haya un INSERT o UPDATE, Se capturan los datos ingresados. En caso contrario (Se realizó un DELETE) no habrá datos nuevos, por lo que se ingresará “Sin datos nuevos”.

**Fecha:** Se registra la fecha y hora exactas en las que se realizó la acción.

parte del sistema en la que se llevó a cabo la acción. Por ejemplo, "Carrito de Ventas".

**SQL Ejecutado:** Se guarda un registro de la consulta SQL ejecutada para realizar el cambio (Ya sea insert, update o delete)

**SQL de Reversa:** Se guarda un registro de la consulta SQL que se debe ejecutar en caso de querer deshacer el cambio (Ya sea insert, update o delete). En caso de INSERT, la consulta contraria es un DELETE, de UPDATE es UPDATE con datos Previos, de DELETE es el INSERT.

• Función de restaurar cambio:  
El administrador ingresa dentro de Administrar -> Auditar Bitacora de Cambios, Selecciona un registro de la grilla (Donde se muestra la bitácora de cambios: Usuario, acción, dato previo, dato posterior y fecha) y hace click en Restaurar Cambio.

El sistema ejecuta la consulta guardada en la base de datos “reverseSQL” la cual revertirá el cambio.

• Base de Datos:

Tabla

Descripción generada automáticamente

Guardar bitácora de Cambios –Dimension Interna

Diagramade secuencia guardar cambios:

Diagrama

Descripción generada automáticamente

Diagramade clases bitácora de cambios:

****

**DER Bitacora de cambio:**

**Tabla

Descripción generada automáticamente**

Auditar bitácora de Cambios – Dimension Externa

**Descomposición Funcional:**

1. El usuario administrador ingresar en Bitácora -> Bitácora Cambios
2. El sistema muestra una grilla con los datos de la bitácora de cambios, una sección especializada para filtrar la tabla (ComboBox y Textbox) y un botón para poder deshacer el cambio
3. El usuario completa los campos para aplicar los filtros y hace click en buscar.
4. El sistema actualiza la grilla según los datos filtrados
5. El administrador selecciona un registro y hace click en restaurar cambio
6. El sistema restaura el cambio y actualiza la grilla

**Diagrama de caso de uso:**



**Especificación de Caso de Uso Auditar bitácora de Cambios:**

|  |
| --- |
| **ID y Nombre: CDU41 – Auditar bitácora de Cambios - Dimension Externa** |
| **Objetivo:** Auditar las actividades de los usuarios en el sistema |
| **Actor principal:** Usuario Administrador |
| **Precondiciones**: |
| **Punto de extensión: -** |
| **Disparador:** El usuario realiza una acción en el sistema, y se desea inspeccionar. |
| **Postcondiciones:** Se registra la bitácora. |
| **Escenario principal:**   1. El administrador ingresa dentro de bitácora -> bitácora de cambios 2. El sistema muestra una grilla con todos los datos de la bitácora de cambios, exceptuando las consultas SQL 3. El administrador ingresa en el combobox el campo por el que desea filtrar, por el textbox el filtro y presiona buscar. 4. Se filtra la grilla con los datos según el filtro indicado. 5. El administrador selecciona un registro en la grilla y presiona en “Restaurar Cambio” 6. El sistema se conecta a la base de datos y deshace el cambio realizado 7. Se notifica al usuario el cambio exitoso |
|
| **Flujos alternativos:**  3.1- El usuario no ingresa ningún filtro, el sistema cancela la operación.  4.1- El filtro ingresado no contiene datos. Se notifica al usuario.  7.1- Hubo un error y se notifica al usuario |

**Diagrama de secuencia Auditar Bitácora de Cambios**



Diagrama de clases Auditar bitácora de cambios



**DER Auditar Bitacora de Cambios**

**Tabla

Descripción generada automáticamente**

# T07 GESTION DE RESPALDOS

Esta gestión se utiliza para administrar las copias de seguridad, esto implica gestionar los backup, los archivos y tener la posibilidad de realizar un restore en caso de ser necesario

El administrador del sistema puede realizar un backup de toma la base de datos y también puede realizar una restauración de toda la base de datos a partir de un archivo de respaldo

## Gestion de respaldos - Backup

**Descomposición funcional:**

1. Al ingresar a la gestión de respaldos, Se visualizan dos botones, uno para realizar un backup y otro para realizar un restore.
2. Al presionar el botón Backup, se muestra una ventana donde se puede seleccionar la ruta para la descarga
3. Se descarga automáticamente un archivo .bak

**DIAGRAMA DE CASO DE USO**



**Especificación funcional CDU49 – Gestion de Respaldos BACKUP**

|  |
| --- |
| **ID y Nombre: CDU49 – Gestion de Respaldos BACKUP** |
| **Objetivo:** Realizar un respaldo o restaurar la base de datos |
| **Actor principal:** Usuario Administrador |  |
| **Precondiciones**: El usuario debe estar logueado con los permisos necesarios En el sistema. |  |
| **Punto de extensión: -** |  |
| **Disparador: Se desde realizar un backup en el sistema.** |  |
| **Postcondiciones: Se realiza un backup en el sistema.** |  |
| **Escenario principal:**   1. El sistema despliega un formulario en donde se encuentran los botones de Backup y restore. 2. El usuario selecciona Backup y se despliega un explorador donde se puede seleccionar la ruta de descarga del archivo 3. El sistema hace el backup de la base de datos y descarga un archivo .bak |  |
|  |
| **Flujos alternativos:**  2.1- El usuario selecciona restore.  3.1- Se produce un error y se notifica al usuario. |  |

**Diagrama de Secuencia gestión de respaldos – BACKUP**



Diagrama de clases Realizar Buckup



## Gestion de respaldos – Restore

**Descomposicion funcional gestion de respaldos restore**

1. Se ingresa a la gestión de respaldos
2. Se visualizan dos botones, uno para realizar un backup y otro para realizar un restore.
3. Al presionar el boton restore, se abre una ventana en donde puedo seleccionar un archivo .bak desde la ruta deseada.
4. Se selecciona un archivo .bak y se presiona aceptar. El sistema informa que se ha restaurado la base de datos.

**Diagrama de caso de uso**



**Especificación funcional CDU49 – Gestion de Respaldos RESTORE**

|  |
| --- |
| **ID y Nombre: CDU49 – Gestion de Respaldos RESTORE** |
| **Objetivo:** Realizar un respaldo o restaurar la base de datos |
| **Actor principal:** Usuario Administrador |  |
| **Precondiciones**: El usuario debe estar logueado con los permisos necesarios En el sistema. |  |
| **Punto de extensión: -** |  |
| **Disparador: Se desde realizar un** restaurar **en el sistema.** |  |
| **Postcondiciones: Se realiza un** restaurar **en el sistema.** |  |
| **Escenario principal:**   1. El sistema despliega un formulario en donde se encuentran los botones de Backup y restore. 2. El usuario selecciona RESTORE y se despliega un explorador donde se puede seleccionar el archivo .bak con el cual realizar el restore 3. El sistema carga todas las tablas de la base de datos con los datos restaurados del archivo .bak 4. El sistema muestra un mensaje de confirmación. |  |
|  |
| **Flujos alternativos:**  3.1- Se produce un error y no se modifica la base de datos  4.1- Se produce un error y se avisa al usuario. |  |

**Diagrama de secuencia realizar restore**



**Diagrama de clases Gestion de respaldos - backups**



# T08. Gestion de dígitos verificadores

• Introducción:

La función de los dígitos verificadores permite al sistema comprobar que los datos almacenados en la base de datos tengan integridad y que no se hayan realizado modificaciones indeseadas en la misma.

De esta forma, se puede detectar si se agregaron, se modificaron o si se quitaron datos de alguna tabla por fuera del sistema.

Al realizar el logueo en el sistema, se comprueba el digito verificador, en caso de que haya un error, se despliega un mensaje de error, si el usuario que intento loguearse no es administrador, se le comunica tener paciencia y comunicarse con un administrador. En caso de que el usuario sea el administrador, se le informa de la gravedad de la situación y que tome las medidas correspondientes.

ESTE CONTROL SE UTILIZA SOBRE LAS TABLAS:

Usuarios, Ventas y Compras.

• Funcionamiento del digito verificador:

En las tablas en las que actuará el dígito verificador, se tendrá una nueva columna, llamada dígito verificador horizontal, en la cual se mantendrá un hash generado con la concatenación de toda la fila.

Este procedimiento se realiza para todas las filas de la tabla, quedando un campo de DigítoHorizontal para cada registro.

Además, tendremos una nueva tabla, llamada Digito Verificador Vertical. Esta nueva tabla tendrá la función de mantener la suma de todos los dígitos horizontales de una tabla, de forma que ante un cambio indeseado en el digito horizontal, podremos verificar que la suma deja de ser correcta.

Esta tabla de digito vertical contendrá como clave primaria el nombre de la tabla a la que hace referencia (Usuarios, Ventas o Compras) y el dígito vertical correspondiente a la tabla.

• Atributos Asociados:

Dígito Verificador Horizontal: Esta es una columna adicional en las tablas de la base de datos en las que se implementa el dígito verificador. Almacena un valor hash generado a partir de la concatenación de todos los valores de una fila en esa tabla.

Dígito Verificador Vertical: Se trata de una tabla separada llamada "Digito Verificador Vertical". Su clave primaria consta del nombre de la tabla a la que hace referencia (por ejemplo, Usuarios, Ventas o Compras) y el dígito vertical correspondiente a esa tabla. Almacena la suma de todos los dígitos horizontales de su tabla asociada.

• Cuando Ocurre:

El proceso de dígito verificador horizontal ocurre cada vez que se inserta, actualiza o elimina un registro en la tabla a la que se aplica esta técnica. Después de realizar una de estas operaciones, se recalcula el valor hash para cada fila afectada y se almacena en la columna "Dígito Verificador Horizontal" correspondiente.

El proceso de dígito verificador vertical ocurre también después de las operaciones de inserción, actualización o eliminación, pero a nivel de la tabla completa. Calcula la suma de todos los dígitos horizontales de la tabla asociada y actualiza el valor en la tabla "Dígito Verificador Vertical" correspondiente.

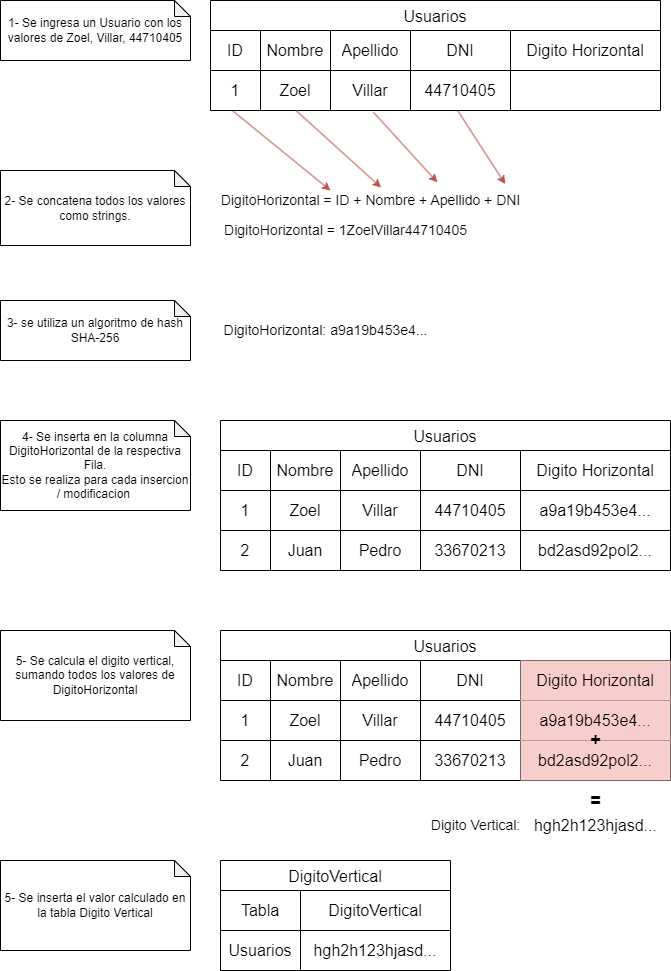
Luego se debe volver a verificar los Dígitos Verificadores al iniciar el sistema, y comprobar si los datos son los correspondiente.

• Método Implementado:

Para el dígito verificador horizontal, se implementa un algoritmo hash que toma todos los valores de una fila en una tabla y los concatena en una cadena. Luego, se aplica una función de hash criptográfico (como SHA-256) a esta cadena para generar un valor hash único. Este valor se almacena en la columna "Dígito Verificador Horizontal".

Para el dígito verificador vertical, se realiza una suma de todos los valores de dígito horizontal en la tabla asociada. Cada vez que se modifica un dígito horizontal en esa tabla, se recalcula la suma y se actualiza en la tabla "Dígito Verificador Vertical" correspondiente.

• Paso a paso del funcionamiento y base de datos:



## Generar Digito Verificador

**Diagrama de secuencia – Generar Digito Verificador**



**Diagrama de clases generar digito verificador**



**DER Generar digito verificador**

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

## Comprobar Digito verificador

Cada vez que el usuario inicie el programa, se ejecutan las funciones que comprueban la integridad del sistema, es decir, se comprueba el digito verificador en las tablas de Usuarios, ventas y compras.

En caso de encontrar un error de integridad, se notifica al usuario que se comunique con un administrador debido a una inconsistencia de datos.

A partir de este momento, el programa deja de permitir a los usuarios hacer Login e iniciar el programa; el único capaz de ingresar es el usuario Administrador, el cual al ingresar se le permitirá realizar un restore de la base de datos o ver en que registro se encontró una inconsistencia de datos.

**Diagrama de Secuencia – Comprobar digito verificador**



**Diagrama de clases – Comprobar digito verificador**



**DER – Comprobar Digito verificador**

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

## Reparar inconsistencias en la base de datos

**Descomposicion funcional reparar inconsistencia en la base de datos.**

1. El administrador ingresa al sistema y realiza el Login con su mail y contraseña
2. El sistema abre únicamente el panel de Solución de Inconsistencias, el cual contiene las opciones de Realizar un Restore y recalcular los dígitos verificadores
3. El usuario presiona calcular dígitos verificadores
4. El sistema muestra la pantalla principal y se solucionan los problemas

**Diagrama de caso de uso – Reparar inconsistencias en la base de datos**

****

**Especificacion de caso de uso - Reparar inconsistencias en la base de datos**

|  |
| --- |
| **ID y Nombre: CDU49 – Gestion de Respaldos RESTORE** |
| **Objetivo:** Realizar un respaldo o restaurar la base de datos |
| **Actor principal:** Usuario Administrador |
| **Precondiciones**: El usuario debe estar logueado con los permisos necesarios En el sistema. |
| **Punto de extensión: -** |
| **Disparador: Se desde realizar un** restaurar **en el sistema.** |
| **Postcondiciones: Se realiza un** restaurar **en el sistema.** |
| **Escenario principal:**   1. El administrador se loguea al sistema y visualiza las opciones de “Realizar Restore” y “Calcular dígitos Verificadores” 2. El administrador selecciona calcular Dígitos Verificadores 3. El sistema vuelve a calcular los dígitos verificadores con los datos que se encuentran en las tablas 4. El sistema notifica que el cálculo se realizó de manera exitosa 5. El sistema muestra la pantalla principal y continúa con normalidad |
|
| **Flujos alternativos:**  2.1- El administrador selecciona realizar restore  2.2- Selecciona la ruta del archivo .bak para realizar el restore.  2.2- El sistema realiza un restore de la base de datos y notifica que se realizó de manera exitosa  2.3- Continua en el punto 5.  2.2.1- Se produce un error realizando el restore y se notifica al administrador  4.1- No se realizó el cálculo de manera exitosa. Se notifica al administrador. |

**Diagrama de secuencia del sistema - Reparar inconsistencias en la base de datos **

**Diagrama de clases - Reparar inconsistencias en la base de datos**



**DER – Reparar inconsistencias en la base de datos**

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente