UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

INGENIERÍA EN SISTEMAS

CICLO IV

ASIGNATURA

Metodología de la Programación

RESPONSABLES

- Alexis David Quizhpe Mendoza
- Danny Michael Jaramillo Jumbo
- Danny Vinicio Vásquez Calderón

DOCENTE

Ing. José Oswaldo Guamán Quinche

FECHA DE PRESENTACIÓN

Domingo 25 de agosto de 2019

LOJA – ECUADOR

2019 – 2020

1 Índice

2 Índice de Figuras

3 Índice de Tablas

4 Título

Sistema para la generación del Kardex y el Libro Diario de la empresa “Zona Wifi”

5 Introducción

Hoy en día los sistemas contables son las bases fundamentales para que la economía de una empresa no decaiga, porque se crean normas o reglas que permitan controlar las operaciones de la empresa, es decir, que brinda muchas ventajas como la generación de reportes, mayor seguridad, así como la administración del dinero y de la información que genera cada operación dentro de la organización. Por ello, el presente proyecto consiste realizar un sistema que le permita al usuario y al administrador de la empresa “Zona Wifi” generar el Kardex de los productos y el Libro Diario de la empresa, aplicando también los requisitos que se obtuvo del administrador cuando se le aplico la encuesta para conocer lo que necesitaba su sistema.

Los datos de las facturas de venta y compra se guardarán en una base de datos, los cuales serán enviados mediante un mapeo de objeto relacional (ORM), que permite crear las tablas en un lenguaje de programación para después convertirlo a lenguaje SQL y levantar las tablas en la base de datos. Además, se utilizará un motor de plantillas para el diseño de las ventanas del proyecto, ayudando a reducir el código, permitiendo reutilizar fragmentos del mismo en otra vista y asegurando que no exista ningún error en el formato de la plantilla que se presentará al usuario.

Por otra parte, se utilizarán librerías y/o plugins que permitan añadir funciones que sean necesarias para completar con los objetivos del proyecto como es el caso de generar y descargar un PDF con la información del Libro Diario y el Kardex de cada producto.

6 Problemática

El libro diario surge de la necesidad de registrar todas y cada una de las operaciones económicas que ocurren dentro de una empresa, ordenándolas de manera cronológica como una evidencia de las acciones que ocurren a nivel financiero, a su vez, para realizar todo este registro se requiere de personas capacitadas en este ámbito como son los contadores, los cuales como cualquier ser humano se equivocan durante dichas anotaciones, todo este proceso conlleva una pérdida de tiempo y dinero. Así mismo, las empresas que utilizan sistemas informáticos para vender sus productos tienen que llevar contabilidad económica, así como registrar las entradas y salidas de los mismos.

Es por ello, que vimos la necesidad de realizar el proceso del libro diario a través de un sitio web, así como, de organizar o llevar el control de manera resumida los ingresos y egresos tanto económicos como de recursos materiales en torno a los servicios o productos ofertados a los usuarios que día a día van adquiriendo o usando en la Empresa “Zona Wifi” que se encuentra ubicada cerca de la Universidad Nacional de Loja, la cual no lleva un registro donde tenga constancia de las ganancias o pérdidas que ocurren diariamente.

7 Objetivo General

- Desarrollar un sistema informático para generar el libro diario de la empresa “Zona Wifi”.

7.1 Objetivos Específicos

- Identificar los datos necesarios para la elaboración del libro diario y el Kardex.
- Registrar los ingresos y egresos de las ganancias y los recursos materiales en un Kardex.
- Generar un documento que presente el reporte del libro diario.

8 Revisión Literaria

8.1 Librerías Utilizadas

Las librerías que se presentaran a continuación fueron utilizadas en el proyecto para facilitar el desarrollo del mismo, permitiendo que el código sea fácil de hacer y de entender.

- **Connect-flash**

Connect-flash es una librería de Express que recibe una solicitud y a la vez esta devolverá un mensaje a la pantalla. (Hanson, 2019)

```
clienteC.saveAll().then(function (result) {
    req.flash('info', 'Se ha registrado correctamente');
    res.redirect('/Clientes');
}).error(function (error) {
    req.flash('error', 'Hubo un problema, comunicate con tu servicio del sistema');
    res.redirect('/Clientes');
});
```

Figura 1 Uso del Connect – Flash

Como se muestra en la *Figura 1*, la librería connect-flash se utilizó en diversas acciones que la persona realizará como son: registrarse como usuario o administrador, guardar, buscar o modificar un producto, entre otras acciones que requieran saber si se ha completado con éxito o no.

- **Nodemailer**

Nodemailer es una librería que permite obtener los datos que ingresa una persona en el formulario de una ventana para después enviar un e-mail a un correo electrónico que haya sido configurado para recibir los mensajes. (Esteibar, 2019)

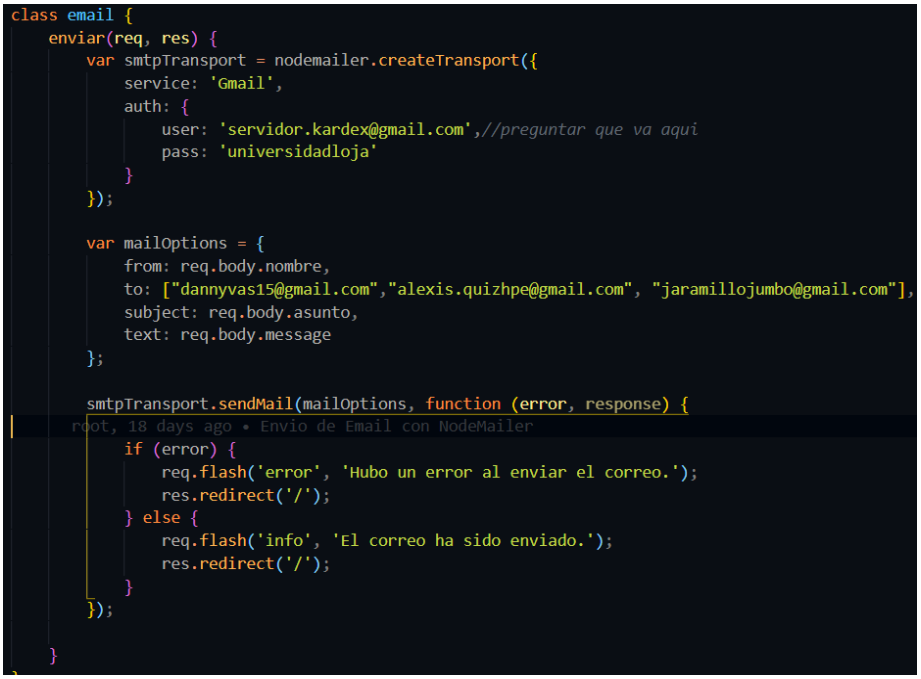


Figura 2 Configuración de Nodemailer

La *Figura 2* muestra la configuración de la librería Nodemailer, definiendo primeramente el correo que enviará el mensaje, para después obtener los datos del formulario en donde se declaran los correos electrónicos los cuales serán los receptores y devolviendo un mensaje a la pantalla confirmando o no el envío del mensaje.

- **Thinky**

Thinky es un Object Relational Mapping (ORM) que transforma las entidades en tablas y convierte los datos en un formato correcto para poder guardar la información en una base de datos. Se lo utilizó para realizar consultas, guardar, obtener y modificar los datos que se registraron en la base de datos. (Michel, 2019)

```
var thinky = require('../config/thinky_init');
var type = thinky.type;
var r = thinky.r;
var Persona = thinky.createModel("Persona", {
    id: type.string(),
    external_id: type.string().default(r.uuid()),
    apellidos: type.string(),
    nombres: type.string(),
    fecha_nac: type.date(),
    edad: type.number(),
    id_rol: type.string(),
    createdAt: type.date().default(r.now())
});
module.exports = Persona;
```

Figura 3 Creación de un modelo usando Thinky

La *Figura 3* demuestra cómo definir un modelo con ayuda del ORM Thinky, estableciendo el tipo de dato para cada atributo y después conectar con la base de datos y crear la tabla “Persona”.

- **RethinkDB**

RethinkDB es una base de datos No SQL utilizada para guardar datos que se envían mediante un lenguaje de programación, los cuales son transformados al lenguaje de la base de datos mediante el ORM Thinky. (Jordana, 2019)

Se utilizó esta base de datos para guardar la información de los productos con sus respectivas ventas y compras, y obtener esos datos para generar un Libro Diario y el Kardex de cada producto.

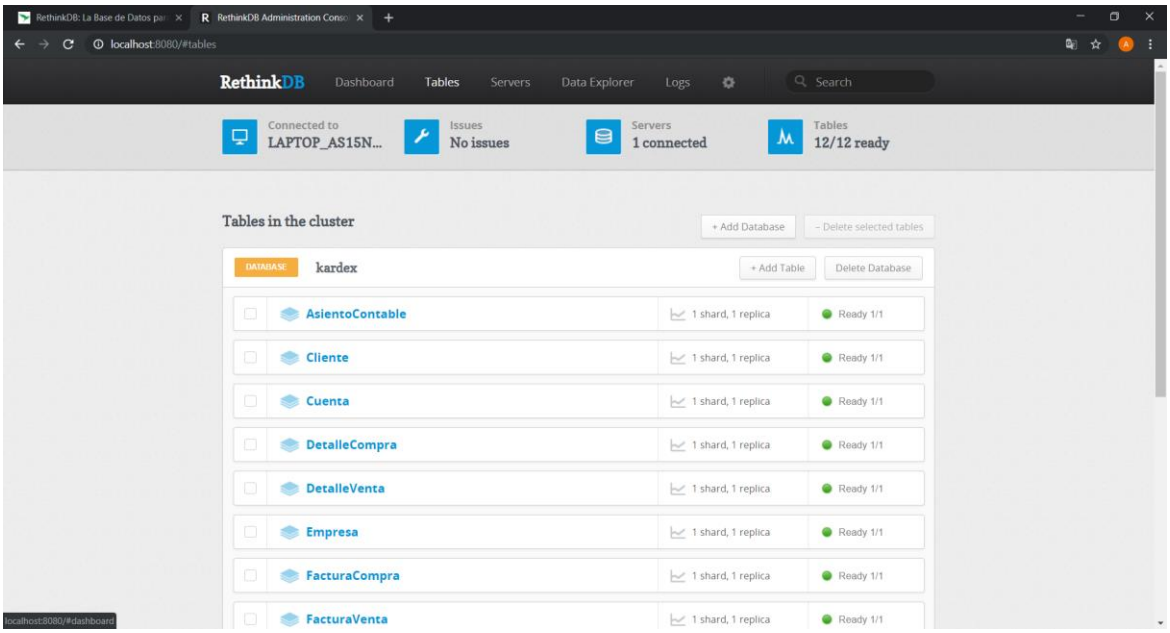


Figura 4 Ventana principal de RethinkDB

La *Figura 4* muestra la interfaz de RethinkDB donde se crea la base de datos con sus respetivas tablas, las cuales fueron creadas con ayuda del ORM Thinky.

- **Express-session**

La librería express-session crea una sesión otorgándole un identificador a la persona cuando inicia sesión para seguir las actividades que tiene permitido realizar, restringiendo el acceso a ventanas a las que no puede acceder sin autorización. (Copes, 2019)

```
if (req.session !== undefined && req.session.datos !== undefined) {
    if (req.session.datos.rol == "Usuario") {
        res.redirect('/usuario');
    } else if (req.session.datos.rol == "Administrador") {
        res.redirect('/Admin');
    }

} else {
    console.log("ROL " + req.session.datos);
    res.render('index', {
        title: 'KARDEX',
        fragmento: 'principal',
        msg: {
            error: req.flash('error'),
            info: req.flash('info')
        }
    });

}
```

Figura 5 Uso del Session para identificar el rol de la persona

La *Figura 5* demuestra el cambio de ventana cada vez que inicie sesión la persona, obteniendo el rol para determinar si es Administrador o Usuario y redirigirlo a la ventana que tiene autorización.

- **Express-handlebars**

Express-handlebars es un motor de plantilla que permite tener más organizado el código, reduciendo el mismo y reutilizando pequeños fragmentos garantizando que no exista ningún error en el formato de la plantilla. Se utilizó el formato express-handlebars (HBS) para las vistas del proyecto. (Katz, 2019)

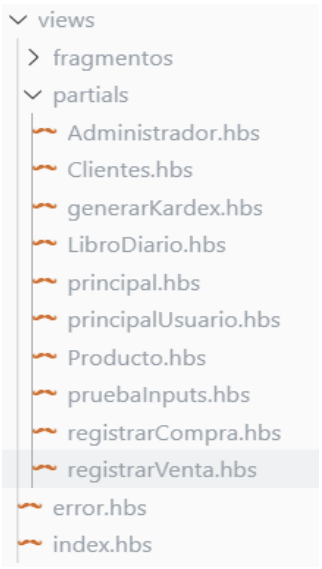


Figura 6 Vistas del proyecto con motor de plantilla HBS

- **JSDoc**

La librería JSDoc permite crear una documentación de la API obteniendo los comentarios en donde se describe la función que realiza cada método declarado en los controladores. (Carmona, 2019). Esta librería la utilizamos para generar la documentación API de nuestra página web.

En la *Figura 7* se muestra en la parte superior del método los comentarios que se generaran describiendo la función que realiza el mismo, los parámetros que recibe y lo que retorno a la vista de la persona.

```
/**
 * @description Metodo para GUARDAR un determinado Cliente
 * @param {req} req  Para tratar los datos del Back end
 * @param {res} res  Parapresentar los datos en  Front end
 * @return Retornamos la vista Clientes
 */
guardar(req, res) {

    cliente.filter({cedula: req.body.cedula2}).then(function (lista){
        if (lista.length >= 1){
            req.flash('error', 'El cliente ya está registrado');
                res.redirect('/Clientes');
        } else {
            var dataP = {
                cedula: req.body.cedula2,
                apellidos: req.body.apellidos2,
                nombres: req.body.nombres2,
                direccion: req.body.direccion2,      You, 3 days ago • Validando Campos de Fact
                telefono: req.body.telefono2
            };
            var clienteC = new cliente(dataP);
            clienteC.saveAll().then(function (result) {
                req.flash('info', 'Se ha registrado correctamente');
                res.redirect('/Clientes');
            }).error(function (error) {
                req.flash('error', 'Hubo un problema, comunicate con tu servicio del sistema');
                res.redirect('/Clientes');
            });
        }
    });
}
```

Figura 7 Descripción de la función y los parámetros del método guardar cliente

Y con la librería JSDoc se genera la API como se muestra a continuación.

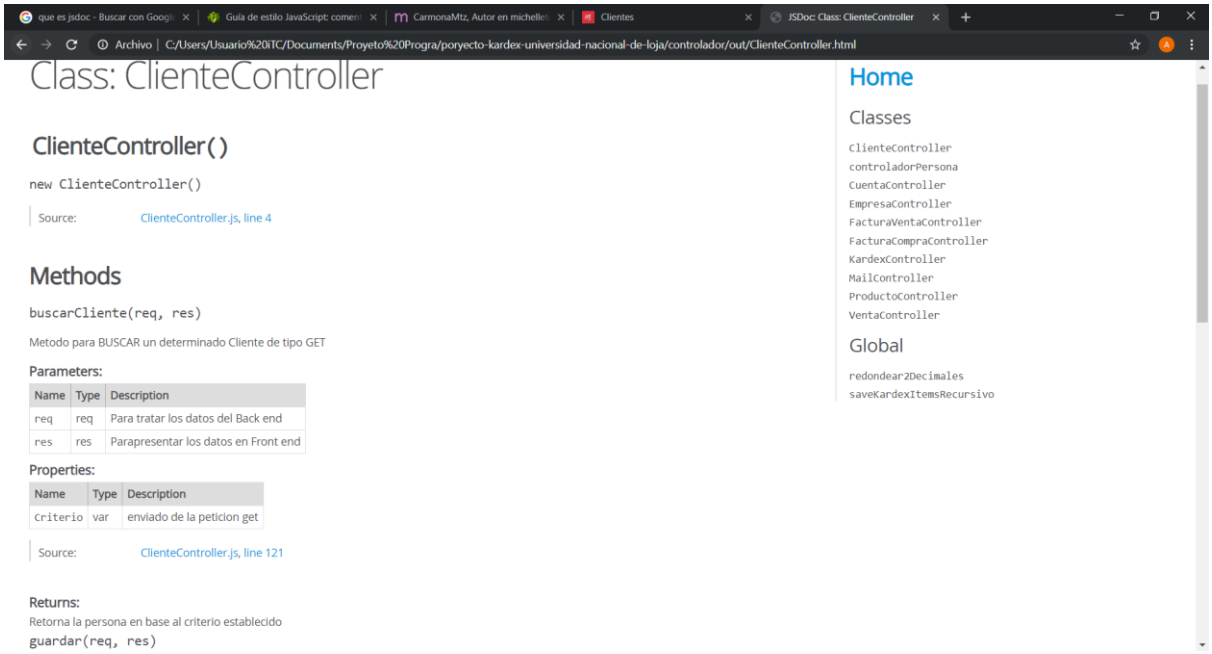


Figura 8 Documentación API del controlador cliente

8.2 Plugins

Por otra parte, los plugins son aplicaciones que se utilizan para agregar una nueva función específica a la aplicación e interactúen entre sí por medio de la interfaz. En el presente proyecto se utilizó únicamente un plugin el cual se menciona a continuación.

- **JSPDF**

JSPDF es una librería que permite al usuario crear archivos en formato PDF a partir de una plantilla HTM de la cual obtiene la información que se está presentando para después generar el PDF. (theeasyprogramming, 2019)

```
getRealStyle: function(elm, attributes, pseudo) {
    var returnObj = {};
    var computed = getComputedStyle(elm, pseudo);
    for(var i=0; i<attributes.length; i++) {
        returnObj[attributes[i]] = computed[attributes[i]];
    }
    return returnObj;
},
copyComputedStyle: function(elm, dest, parentStyle, attributes, pseudo) {
    parentStyle = parentStyle || {};
    var s = xepOnline.Formatter.getRealStyle(elm, attributes, pseudo);

    for ( var i in s ) {
        var currentCss = s[i];

        // ignore duplicate "inheritable" properties
        if(parentStyle !== undefined && (parentStyle[i] && parentStyle[i] === currentCss)) { } else {
            // The try is for setter only properties
            try {
                dest.style[i] = s[i];
                // `fontSize` comes before `font` If `font` is empty, `fontSize` gets
                // overwritten.  So make sure to reset this property. (hackyhackhack)
                // Other properties may need similar treatment
                if ( i == "font" ) {
                    dest.style.fontSize = s.fontSize;
                }
            } catch (e) {}
        }
    }
},
setSVGHeightWidth: function(dest) {
    jQuery(dest).find('svg').each(function(index) {
        var svg = jQuery(this);
        svg.attr('height',svg.outerHeight());
        svg.attr('width',svg.outerWidth());
    });
```

Figura 9 Librería JSDPF

9 Análisis de Requerimientos

Los siguientes requisitos fueron levantados en base a la entrevista tanto al dueño del local como a la persona que le administra la contabilidad de la empresa.

Tabla 1 Requerimientos del Sistema

Referencia	Descripción	Categoría
RF001	Iniciar sesión	E
RF002	Registrar producto	E
RF003	Modificar producto	E
RF004	Registro de transacción en libro diario	E
RF005	Modificar la transacción en libro diario (hasta un día).	E
RF006	Registro de venta	E
RF007	Modificar la venta (un día)	E
RF008	Registro de compra	E
RF009	Modificar la compra (un día)	E
RF010	Registrar devoluciones de productos (extra)	E
RF011	Modificar devoluciones de productos(extra)	E
RF012	Notificar si un producto tiene mucha demanda	E
RF013	Notificar si un producto tiene poca demanda	E
RF014	Notificar cuando un producto este en su punto crítico	E
RF015	Actualizar kardex de cada transacción registrada en el libro diario	E

RF016	Observar lista de productos	E
RF017	Generar un kardex descargable en formato PDF	E
RF018	Generar el libro diario descargable en formato PDF.	E

9.1 Casos de Uso

Tabla 2 Inicio de Sesión

Identificación del caso de uso:	UC001
Nombre del caso de uso:	Iniciar sesión
Características:	RF001: Inicio de sesión
Descripción del requerimiento:	Se podrán acceder al sistema en base a un usuario y contraseña, y se mostrara los menús que estén autorizados para el mismo.
Prioridad del requerimiento: Alto	

Tabla 3 Administrar producto

Identificación del caso de uso:	UC002
Nombre del caso de uso:	Administrar producto.
Características:	RF002: Registrar producto. RF003: Modificar producto.
Descripción del requerimiento:	Debe ser validada cada parámetro de manera interna y en el caso de que no sean, se deberá mostrar una alerta. Solo funciona en el caso del administrador.
Prioridad del requerimiento: Alto	

Tabla 4 Administrar transacciones en el libro diario

Identificación del caso de uso:	UC003
Nombre del caso de uso:	Administrar transacciones en el Libro Diario.
Características:	RF004: Registro de transacción en libro diario. RF005: Modificar la transacción en libro diario.
Descripción del requerimiento:	Debe ser validada cada parámetro de manera interna y en el caso de que no sean, se deberá mostrar una alerta que podrá observar el usuario.
Prioridad del requerimiento: Alto	

Tabla 5 Administrar la venta de productos

Identificación del caso de uso:	UC004
Nombre del caso de uso:	Administrar la venta de productos
Características:	RF006: Registrar la venta RF007: Modificar la venta.
Descripción del requerimiento:	Se realizará un CRUD de las ventas, en el caso de modificar y dar de baja, solo contará con un día para realizar estas dos operaciones.
Prioridad del requerimiento: Alto	

Tabla 6 Administrar la compra de productos

Identificación del caso de uso:	UC005
Nombre del caso de uso:	Administrar la compra de productos
Características:	RF008: Registrar la compra. RF009: Modificar la compra.
Descripción del requerimiento:	Se realizará un CRUD de las compras, en el caso de modificar y dar de baja, solo contará con un día para realizar estas dos operaciones.
Prioridad del requerimiento: Alto	

Tabla 7 Administrar las devoluciones

Identificación del caso de uso:	UC006
Nombre del caso de uso:	Administrar las devoluciones
Características:	RF010: Registrar las devoluciones de productos. RF011: Modificar las devoluciones de productos.
Descripción del requerimiento:	Se realizará un CRUD de las devoluciones de los diferentes productos.
Prioridad del requerimiento: Baja	

Tabla 8 Notificaciones para el usuario

Identificación del caso de uso:	UC007
Nombre del caso de uso:	Notificaciones para el usuario
Características:	RF012: Notificar si un producto tiene mucha demanda RF013: Notificar si un producto tiene poca demanda RF014: Notificar cuando un producto este en su punto crítico
Descripción del requerimiento:	Advertir al usuario cuando ocurra cualquiera de estos casos de uso.
Prioridad del requerimiento: Baja	

Tabla 9 Actualización de kardex

Identificación del caso de uso:	UC008
Nombre del caso de uso:	Actualización de kardex
Características:	RF015: Actualizar kardex de cada transacción registrada en el libro diario .
Descripción del requerimiento:	El sistema deberá actualizar de forma automática el kardex de cada producto según se actualice las transacciones en el libro diario.
Prioridad del requerimiento: Alto	

Tabla 10 Reporte de Productos

Identificación del caso de uso:	UC009
Nombre del caso de uso:	Reporte de Productos
Características:	RF016: Observar un reporte de los productos existentes.
Descripción del requerimiento:	El usuario y administrador tendrá la facilidad de observar un reporte de los productos existentes.
Prioridad del requerimiento: Bajo	

Tabla 11 Descargar Kardex

Identificación del caso de uso:	UC010
Nombre del caso de uso:	Descargar Kardex.
Características:	RF017: Generar un kardex descargable en formato PDF
Descripción del requerimiento:	El usuario y administrador tendrá la facilidad de descargar el kardex de un producto específico.
Prioridad del requerimiento: Bajo	

Tabla 12 Reporte de Libro Diario

Identificación del caso de uso:	UC011
Nombre del caso de uso:	Reporte de Libro Diario
Características:	RF018: Generar el libro diario descargable en formato PDF
Descripción del requerimiento:	Generar el libro diario en formato PDF observable y descargable por el usuario y el administrador.
Prioridad del requerimiento: Bajo	

10 Diagrama de Casos de Uso

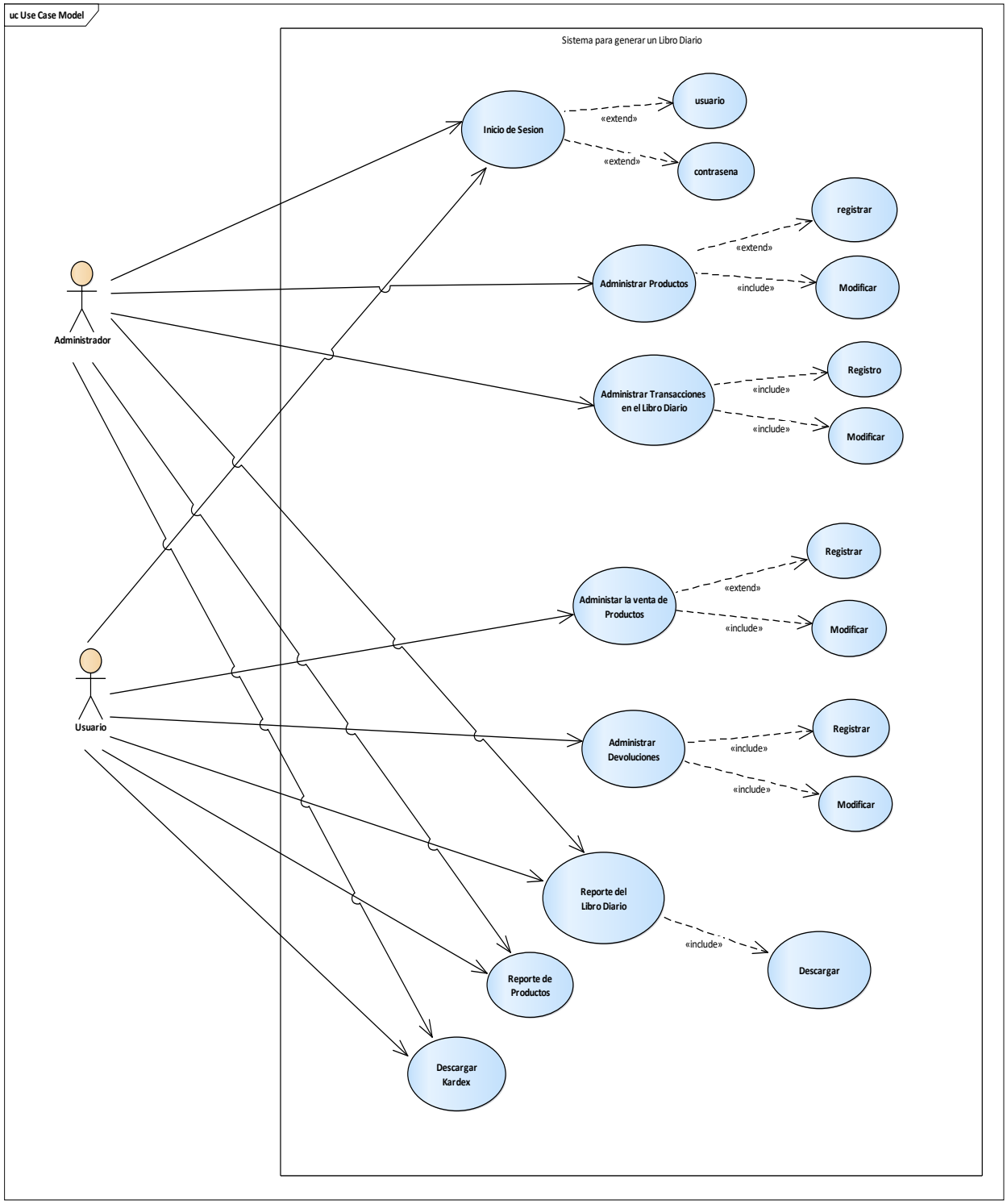


Figura 10 Diagrama de casos de uso

11 Diagrama de Clases

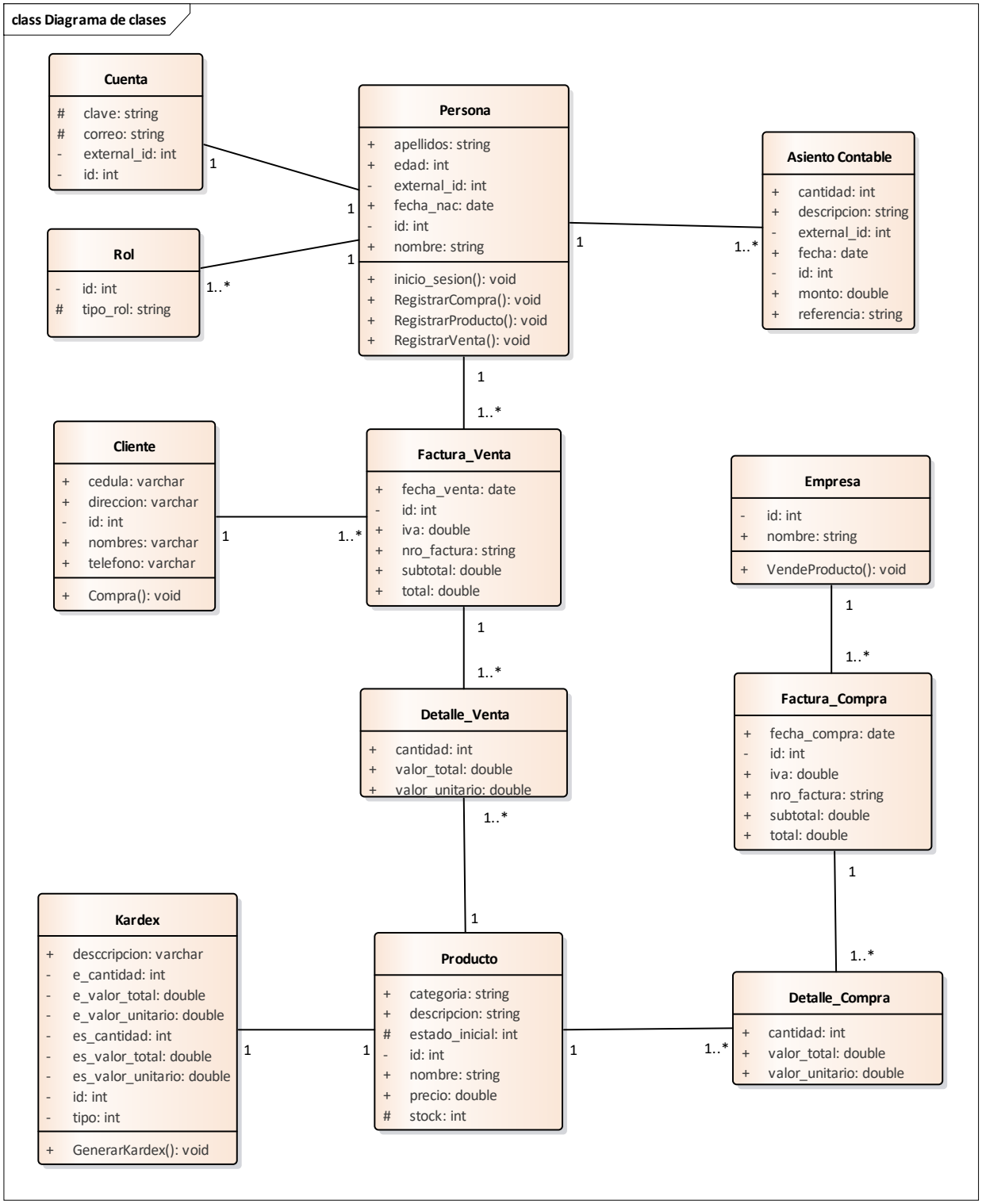


Figura 11 Diagrama de Clases

12 Diagrama de la Base de Datos

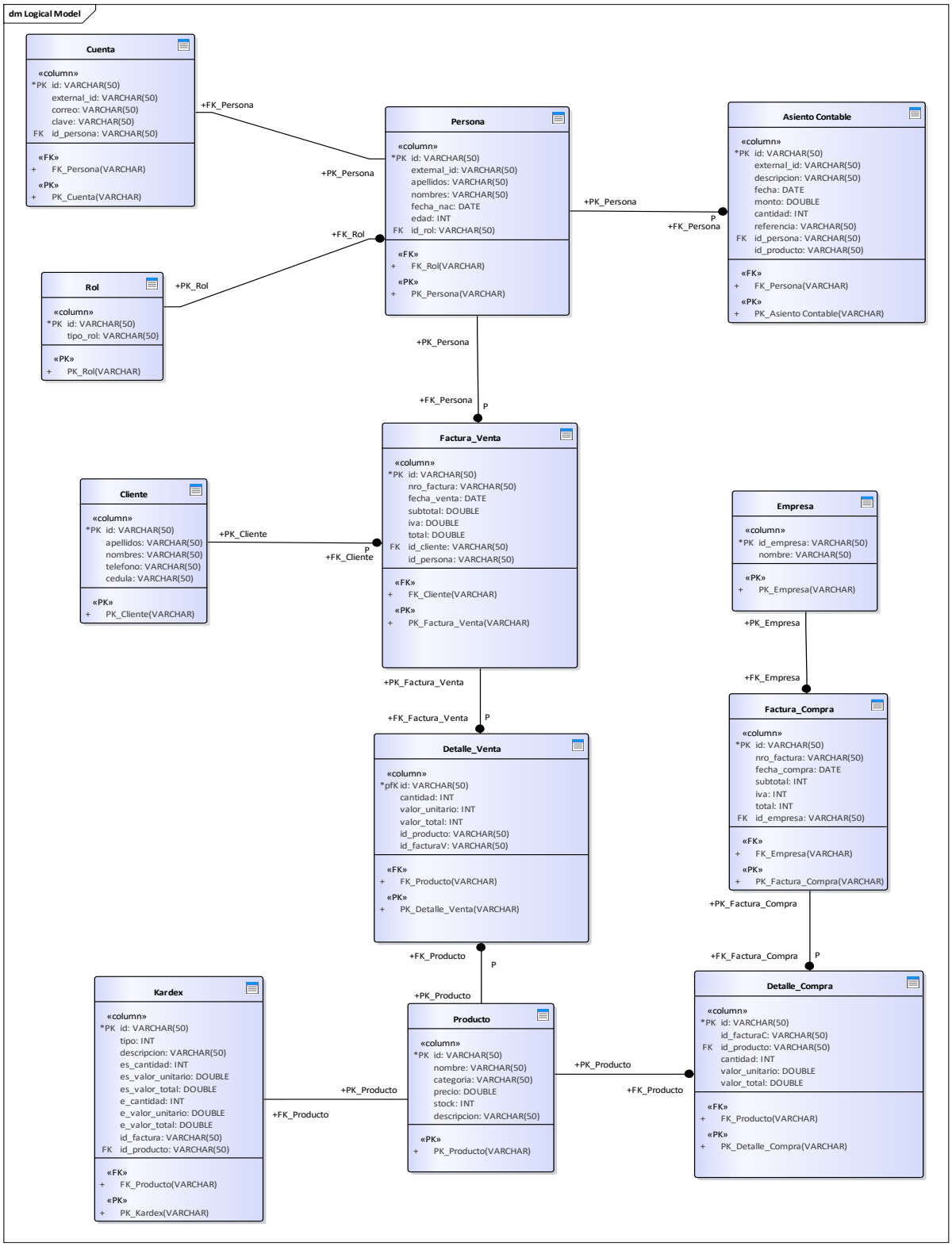


Figura 12 Diagrama de la Base de Datos

13 Resultados

13.1 Identificar los datos necesarios para la elaboración del libro diario y el Kardex

Los datos necesarios para poder generar la información del Libro Diario y del Kardex se la tuvo que consultar en diferentes páginas que muestren ejemplos con la estructura más importante de los mismos.

La estructura del Kardex contiene información de las facturas de compra y venta que se realizaron, de las cuales se obtiene la fecha de la factura, el detalle de si fue compra o venta, la cantidad de productos que se compró o se vendió, el valor unitario del producto y la cantidad total al que fue comprado o vendido, ubicando la información de compra en la sección de entrada y la información de la venta en la sección de salidas como se muestra a continuación.

KÀRDEX

Artículo:			lavadoras			Existencia mínima:			60		
Método:			Promedio ponderado			Excistencia máxima:			495		
Fecha			Detalle	Entradas			Salidas			Existencias		
D	M	A		Cantidad	V/ Unitario	V/ Total	Cantidad	V/ Unitario	V/ Total	Cantidad	V/ Unitario	V/ Total
3	5	11	Saldo anterior							98	94	9212
5	5	11	Compra según factura Nº20	18	134	2412				116	100,21	11624
7	5	11	Venta según factura Nº01				67	100,21	6714,07	49	100,2	4909,93
9	5	11	Venta según factura Nº02				17	100,2	1703,4	32	100,2	3206,53
11	5	11	Compra según factura Nº35	95	135	12825				127	126,23	16031,5
			Inventario Final							127	126,23	16031,5

Figura 13 Kardex

Obteniendo esos atributos se pudo generar el Kardex y así calcular las existencias totales del producto para conocer su valor unitario, la cantidad total que aún se tiene disponible y el valor total que conforman todos ellos.

Para generar el Libro Diario, se obtuvo la información de las facturas de venta y compra seleccionando la fecha de la factura, el valor total que se registra en la factura, el número de la factura como referencia y en la descripción el detalle de si fue compra o venta como se muestra a continuación.

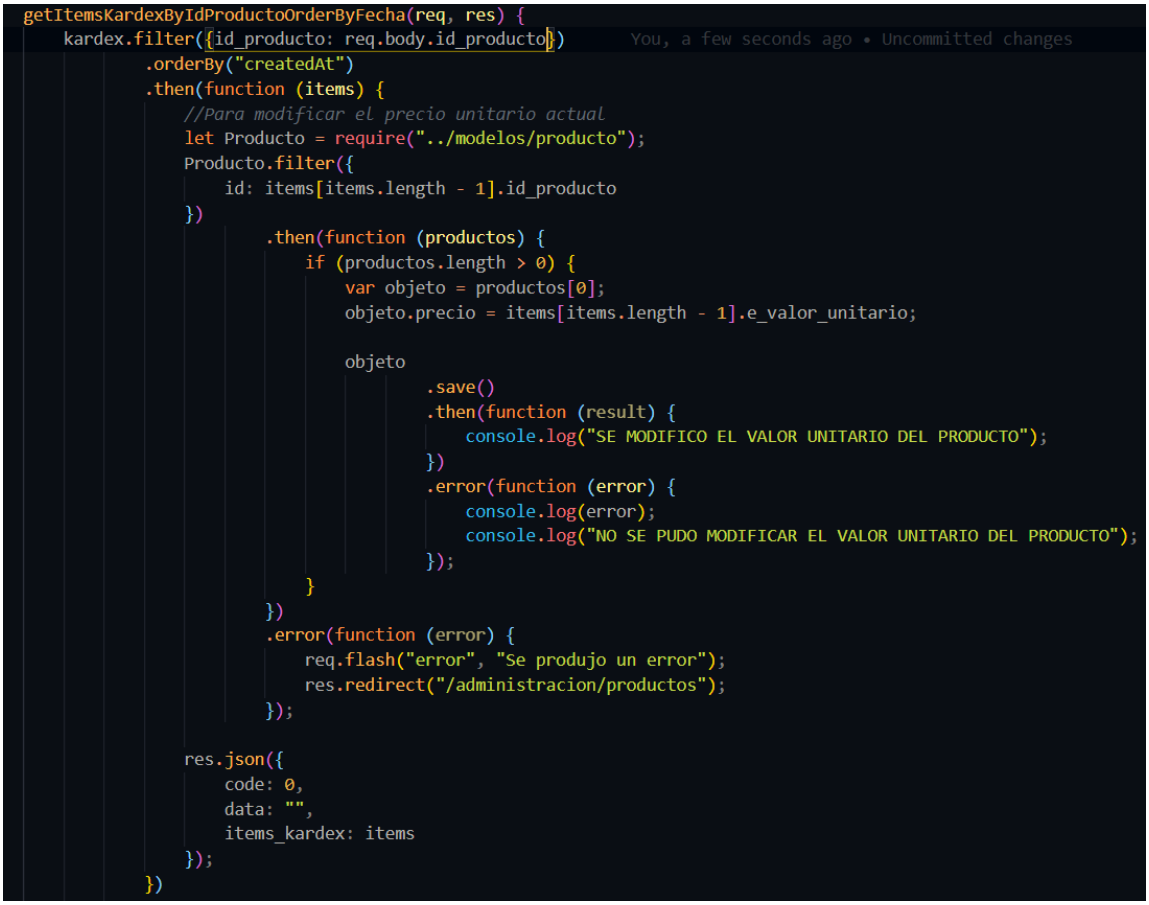
Libro diario de las Transacciones				
Fecha	**Descripcion**	**Referencia**	**Debe**	**Haber**
20-Agos	Compra con una cantidad de 100 unidades, por un monto de : $ 200 Caja	FAC 0000000001 ---	$ 200 ---	--- $ 200
20-Agos	Caja Venta con una cantidad de 1 unidades, por un monto de : $ 1.64	--- FAC 0000000004	$ 1.64 ---	--- $ 1.64
20-Agos	Caja Venta con una cantidad de 13 unidades, por un monto de : $ 65	--- FAC 0000000003	$ 65 ---	--- $ 65
20-Agos	Compra con una cantidad de 100 unidades, por un monto de : $ 500 Caja	FAC 0000000002 ---	$ 500 ---	--- $ 500
20-Agos	Caja Venta con una cantidad de 10 unidades, por un monto de : $ 50	--- FAC 0000000002	$ 50 ---	--- $ 50
20-Agos	Compra con una cantidad de 50 unidades, por un monto de : $ 50 Caja	FAC 0000000003 ---	$ 50 ---	--- $ 50
20-Agos	Caja Venta con una cantidad de 10 unidades, por un monto de : $ 20	--- FAC 0000000001	$ 20 ---	--- $ 20
----	**SUMAN**	**----**	**$ 886.64**	**$ 886.64**

Figura 14 Libro Diario

13.2 Registrar los ingresos y egresos de las ganancias y los recursos materiales en un Kardex

Como se observa en la *Figura 11,* el Kardex contiene información que se registra en las facturas de compra y venta.

Para generar el Kardex se tuvo que realizar consultas a la base de datos obteniendo la cantidad que se vendió o se compró del producto en cada factura, el valor unitario, la fecha de cada factura y el detalle de sí esta es compra o venta.



```
getItemsKardexByIdProductoOrderByFecha(req, res) {
    kardex.filter({id_producto: req.body.id_producto})      You, a few seconds ago • Uncommitted changes
          .orderBy("createdAt")
          .then(function (items) {
              //Para modificar el precio unitario actual
              let Producto = require("../modelos/producto");
              Producto.filter({
                  id: items[items.length - 1].id_producto
              })
                      .then(function (productos) {
                          if (productos.length > 0) {
                              var objeto = productos[0];
                              objeto.precio = items[items.length - 1].e_valor_unitario;

                              objeto
                                      .save()
                                      .then(function (result) {
                                          console.log("SE MODIFICO EL VALOR UNITARIO DEL PRODUCTO");
                                      })
                                      .error(function (error) {
                                          console.log(error);
                                          console.log("NO SE PUDO MODIFICAR EL VALOR UNITARIO DEL PRODUCTO");
                                      });
                          }
                      })
                      .error(function (error) {
                          req.flash("error", "Se produjo un error");
                          res.redirect("/administracion/productos");
                      });

              res.json({
                  code: 0,
                  data: "",
                  items_kardex: items
              });
          })
```

Figura 15 Consulta para generar el Kardex de un producto

Para calcular los datos de la sección existencial del Kardex se realizaron las siguientes operaciones:

- Cuando es compra

1. Se multiplica las cantidades compradas por el valor unitario del producto para obtener el valor total de la compra en la sección de “Entradas”.
2. Se multiplica las cantidades existentes por el valor unitario del producto para obtener el valor total del producto existente en la sección de “Existencias”.
3. Se suma la cantidad de “Entradas” más la cantidad de “Existencias” y después se suma el valor total de “Entrada” más el valor total de “Existencias”.

4. Finalmente, se divide el valor total de la suma de “Entradas” y “Existencias” para la cantidad total de la suma entre de “Entradas” y “Existencias” y así obtener el precio unitario de la sección de “Existencias”.

- Cuando es venta

1. Se multiplica las cantidades vendidas por el valor unitario del producto para obtener el valor total de la venta en la sección de “Salidas”.
2. Se multiplica las cantidades existentes por el valor unitario del producto para obtener el valor total del producto existente en la sección de “Existencias”.
3. Se resta la cantidad de “Salidas” menos la cantidad de “Existencias” para luego restar el valor total de “Salidas” menos el valor total de “Existencias”.
4. Finalmente, se divide el valor total de la resta de “Entradas” y “Existencias” para la cantidad total de la resta entre de “Entradas” y “Existencias” para así obtener el valor unitario de la sección de “Existencias”.

Estas operaciones se realizan siempre con los detalles de la factura anterior. En caso de que no exista, los datos de la primera factura ingresada pasaran directamente a la sección de “Existencias” de la cual partirán todas las operaciones para generar el Kardex mediante el método de promedio ponderado.

13.3 Generar un documento que presente el reporte del libro diario

Para generar un documento que contenga la información generada del Libro Diario se usó un plugin llamado JSPDF (*Figura 7*), el cual importa todo lo que está en el *body* de la página HTML.

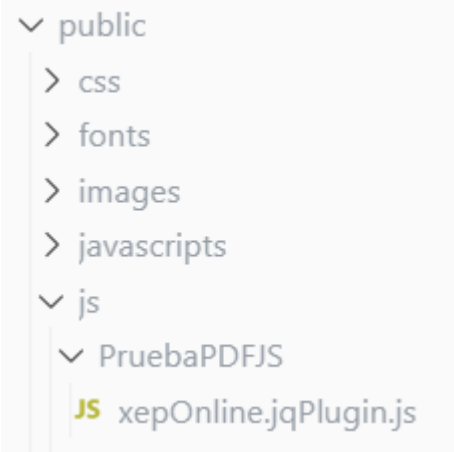


Figura 16 Ubicación del plugin JSPDF en el proyecto

Como se observa en la *Figura 14*, el archivo donde está configurado el plugin para generar los documentos en formato PDF se encuentra en la carpeta *Public* del proyecto debido a que es un archivo javascript. Para generar el documento se llama a la función *xepOnline.Formatter.Format* en una etiqueta de tipo enlace donde se le da la condición para que cada vez que se haga click sobre la etiqueta se retornar un archivo PDF con el contenido de la vista.

```
<div class="col b">
    <a id="exportar_pdf"
       onclick="return xepOnline.Formatter.Format('item_detalleVenta',
       {srctype: 'html', render: 'download', filename: 'Prueba de kardex', cssStyle: [{padding: '0'}]});">PDF</a>
</div>
```

Figura 17 Etiqueta para generar el PDF

14 Conclusiones

Finalmente, al haber culminado con el proyecto y cumpliendo con los objetivos general y específicos se puede concluir lo siguiente:

- Los atributos establecidos en el asiento contable, las facturas de venta y compra, y en los detalles de venta y compra, fueron los indicados y necesarios para generar el Kardex de cada producto y el Libro Diario de la empresa.
- Se logró generar el Kardex de cada producto, obteniendo los valores de las ventas y compras mediante consultas hechas a la base de datos para luego calcular su resultado por medio del método del promedio ponderado.
- La librería JSPDF facilito la opción de generar el Libro Diario con toda la información de las facturas de venta y compra, y el Kardex de los productos en un archivo con formato PDF descargable.

15 Recomendaciones

A continuación, se desea sugerir algunas recomendaciones cuya implementación pueden facilitar y mejorar el desarrollo de su proyecto.

- Utilizar motores de plantilla para reutilizar código, además de que permiten separar la presentación de la lógica de la aplicación en la que se está trabajando, es favorable para que esta no sea tan pesada.
- Utilizar plantillas HTML5 ya codificadas si no se desea crear una propia, así se evitará perdida innecesaria de tiempo y si la implementa en un motor de plantilla puede aprovechar su diseño al máximo.
- Utilizar librerías y plugins para agregarle diversas funcionalidades y efectos a las páginas de internet, permitiendo así manipular la estructura de las páginas de forma dinámica y el estilo visual de cada una.

16 Bibliografía

Carmona. (23 de Agosto de 2019). *michelletorres*. Obtenido de Documentación de API (Javascript) con JSDoc: https://blog.michelletorres.mx/documentacion-de-api-javascript-con-jsdoc/

Copes, F. (20 de Agosto de 2019). *Express Sessions*. Obtenido de Cómo usar sesiones para identificar usuarios en todas las solicitudes: https://flaviocopes.com/express-sessions/

Esteibar, U. (19 de Agosto de 2019). *Medium*. Obtenido de Envía emails desde Node.js con Nodemailer: https://medium.com/@uesteibar/env%C3%ADa-emails-desde-node-js-con-nodemailer-178cacf5cf6b

Hanson, J. (19 de Agosto de 2019). *Connect-Flash*. Obtenido de Middleware de mensajes flash para Connect y Express.: https://github.com/jaredhanson/connect-flash

Jordana, J. (20 de Agosto de 2019). *Openexpoeurope*. Obtenido de RethinkDB: La Base de Datos para la Web en Tiempo Real: https://openexpoeurope.com/es/rethinkdb-bbdd-para-web-tiempo-real/

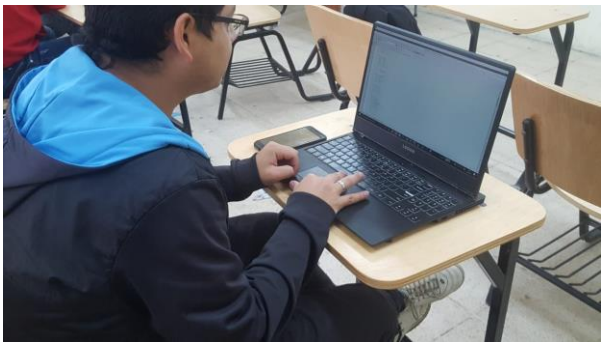
Katz, Y. (20 de Agosto de 2019). *GitHub*. Obtenido de Handlebars.js: https://github.com/wycats/handlebars.js/

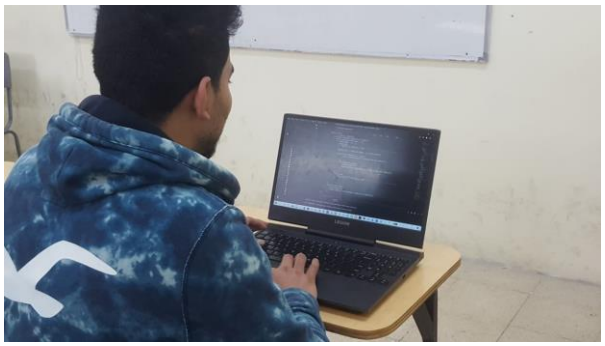
Michel. (20 de Agosto de 2019). *Justonepixel.com*. Obtenido de Thinky: http://justonepixel.com/thinky/

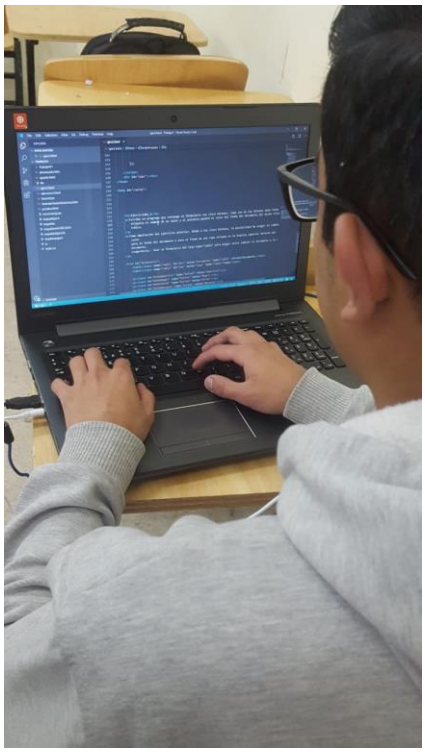
Ruben. (22 de Agosto de 2019). *TusEjemplos*. Obtenido de Ejemplos de kardex: https://tusejemplos.com/ejemplos-de-kardex/

theeasyprogramming. (2019 de Agosto de 2019). *Easy programming*. Obtenido de jsPDF – Opción sencilla para creación de reportes: https://theeasyprogramming.wordpress.com/author/theeasyprogramming/

17 Anexos







Preguntas aplicadas en la encuesta para obtener los Requisitos del Sistema

1. **¿Desea que el sistema genere un reporte sobre los ingresos y egresos?**

Si

2. **¿Desea tener una medida de seguridad que impida que otros usuarios realicen su trabajo como administrador?**

Si

3. **¿Qué sistemas operativos utiliza?**

Únicamente Windows

4. **¿Desea una interfaz llamativa o con colores de baja iluminación?**

No, una interfaz simple.

5. **¿Desea recibir notificaciones sobre alguna acción necesario?**

Cuando un producto tenga un stock bajo.

6. **¿Desea tener animaciones o que cumpla el objetivo principal?**

Sin animaciones, que se centre en cumplir el objetivo principal.