**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA**

**SEDE VIÑA DEL MAR – JOSÉ MIGUEL CARRERA**

# SISTEMA DE ADMINISTRACIÓN DE CIBERCAFÉ “INTERPLAY”

Trabajo de Titulación para optar al Título de Técnico Universitario en INFORMÁTICA

Alumnos:

Sergio Esteban Mora Berrueta

Andrés Ignacio Quezada Reyes

Profesor Guía:

Srta. Lais San Martín Navarro

**2015**

# RESUMEN

**KEYWORDS:** SISTEMA DE CONTROL DE CIBERCAFÉ, VENTAS Y CONTROL DE STOCK.

El presente texto tratará sobre un sistema de información desarrollado para el cibercafé InterPlay, ubicado en Villa Rukán, Reñaca Alto, Viña del Mar, que busca agilizar el proceso de venta de artículos, que es realizado, actualmente, de forma manual. También, se quiere arreglar algunas dificultades técnicas del software utilizado hoy en día por el cibercafé para controlar el uso de lo equipos disponibles (que pueden ser computador o consola de videojuegos). Aunado a esto, el sistema busca llevar un registro claro de los precios y cantidades de los artículos poseídos por el cibercafé.

En el capítulo 1, se explica la situación actual del cibercafé, cómo opera normalmente, quiénes son los actores involucrados, qué falencias y/o imperfecciones fueron detectadas, la solución que se propone, cuáles son los objetivos, las metas, de este proyecto, cómo se estructurará el nuevo sistema tal que optimice los procesos, cómo será el modelo de datos relacional del sistema, qué funcionalidades serán incluidas y mejoradas y las tecnologías sobre las cuales se construirá el sistema.

En el capítulo 2 se describe el ambiente computacional en que el sistema será insertado. Esto incluye el hardware (identificación de procesadores, almacenamiento primario y secundario, puertos de entrada y salida), software (qué sistema(s) operativo(s) será(n) soportado(s), qué herramientas de programación se utilizará, qué sistema de gestor de base de datos será utilizado y una breve explicación de cada uno de éstos). También, describirá el modelo de datos (relacional) del sistema. Primero, habrá una descripción general, listando cada una de las tablas que la conforman, con una pequeña descripción de cuál es la funcionalidad de cada una de ellas, para luego detallar cada campo y clave para cada una de las tablas en cuestión.

En el capítulo 3, se describen los programas que componen el sistema. Primero, se da una mirada general, desde un punto de vista de usuario, a través del diagrama de menú, que muestra cómo estará compuesto el menú que navegará el usuario. Luego, desde una perspectiva algorítmica, se presentan todos las partes componentes del sistema, de manera resumida, en el diagrama modular. Seguido a esto, se describen los programas en detalle en la descripción de programas.

Finalmente se presentarán las conclusiones del trabajo.

# ÍNDICE

[RESUMEN](#_Toc427870275)

[INTRODUCCIÓN](#_Toc427870280)

[CAPÍTULO 1: ASPECTOS RELEVANTES DEL DISEÑO LÓGICO](#_Toc427870281)

[1.1. DESCRIPCIÓN DE LA ORGANIZACIÓN](#_Toc427870282)

[1.2. DESCRIPCIÓN DE LA SITUACIÓN ACTUAL](#_Toc427870283)

[1.3. PROBLEMAS DETECTADOS](#_Toc427870284)

[1.4. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA](#_Toc427870285)

[1.4.1. Objetivo principal](#_Toc427870286)

[1.4.2. Objetivos específicos](#_Toc427870287)

[1.4.3. Beneficios del sistema](#_Toc427870288)

[1.4.4. Descripción general de la solución propuesta](#_Toc427870289)

[1.4.5. Diagrama de flujo administrativo para el sistema propuesto](#_Toc427870290)

[1.4.6. Estructura funcional del sistema](#_Toc427870291)

[1.4.7. Detalle de las funcionalidades por cada proceso](#_Toc427870292)

[1.4.8. Manejo de información](#_Toc427870293)

[1.4.9. Estructura de códigos](#_Toc427870294)

[1.4.10. Condicionantes de diseño](#_Toc427870295)

[CAPÍTULO 2: MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS](#_Toc427870296)

[2.1. DESCRIPCIÓN DEL RECURSO COMPUTACIONAL](#_Toc427870297)

[2.1.1. Configuración del sistema](#_Toc427870298)

[2.1.2. Software utilizado](#_Toc427870299)

[2.2. DESCRIPCIÓN DE ARCHIVOS](#_Toc427870300)

[2.2.1. Listado de tablas](#_Toc427870301)

[2.2.2. Codificaciones](#_Toc427870302)

[2.2.3. Tipos de datos](#_Toc427870303)

[2.2.4. Descripción de archivos](#_Toc427870304)

[CAPÍTULO 3: DESCRIPCIÓN DE PROGRAMAS](#_Toc427870305)

[3.1. DIAGRAMA DE MENU](#_Toc427870306)

[3.2. DIAGRAMA MODULAR](#_Toc427870307)

[3.3. LISTADO DE PROGRAMAS](#_Toc427870308)

[3.4. DISEÑO DE PROGRAMAS](#_Toc427870309)

[3.4.1. Login](#_Toc427870310)

[3.4.2. Menú principal](#_Toc427870311)

[3.4.3. Venta de productos](#_Toc427870312)

[3.4.4. Cuadrar caja](#_Toc427870313)

[3.4.5. Ingreso de factura](#_Toc427870314)

[3.4.6. Mantenedor de trabajadores](#_Toc427870315)

[3.4.7. Informe de productos](#_Toc427870316)

[3.4.8. Informe de stock crítico](#_Toc427870317)

[3.4.9. Arriendos](#_Toc427870318)

[3.4.10. Arrendar](#_Toc427870319)

[CONCLUSIONES Y RECOMENDACIONES](#_Toc427870320)

[BIBLIOGRAFÍA](#_Toc427870321)

[ANEXO A: ARCHIVOS DE PROGRAMA](#_Toc427870322)

# ÍNDICE DE FIGURAS

[Figura 1-1. Modelo de datos relacional “Sistema de Administración de cibercafé”](#f_1_1)

[Figura 3-1. Diseño de pantalla login](#f_3_1)

[Figura 3-2. Diseño de pantalla menú principal](#f_3_2)

[Figura 3-3. Diseño de pantalla venta](#f_3_3)

[Figura 3-4. Diseño de pantalla cuadrar caja](#f_3_4)

[Figura 3-5. Diseño de pantalla de ingreso de factura](#f_3_5)

[Figura 3-6. Diseño de pantalla mantenedor de trabajadores](#f_3_6)

[Figura 3-7. Diseño de pantalla de informe de productos](#f_3_7)

[Figura 3-8. Diseño de pantalla de informe de stock crítico](#f_3_8)

[Figura 3-9. Diseño de pantalla arriendos](#f_3_9)

[Figura 3-10. Diseño de pantalla arrendar](#f_3_10)

# ÍNDICE DE TABLAS

[Tabla 2-1. Listado de tablas](#t_2_1)

[Tabla 2-2. Tipos de datos numéricos](#t_2_2)

[Tabla 2-3. Tipos de datos fecha y hora](#t_2_3)

[Tabla 2-4. Tipos de datos de cadena de caracteres](#t_2_4)

[Tabla 2-5. Tabla usuario](#t_2_5)

[Tabla 2-6. Tabla tiempos en reserva](#t_2_6)

[Tabla 2-7. Tabla historial sesiones trabajador](#t_2_7)

[Tabla 2-8. Tabla venta](#t_2_8)

[Tabla 2-9. Tabla venta anulada](#t_2_9)

[Tabla 2-10. Tabla detalle venta](#t_2_10)

[Tabla 2-11. Tabla producto](#t_2_11)

[Tabla 2-12. Tabla categoría](#t_2_12)

[Tabla 2-13. Tabla detalle\_fact](#t_2_13)

[Tabla 2-14. Tabla factura](#t_2_14)

[Tabla 2-15. Tabla historial precio](#t_2_15)

[Tabla 2-16. Tabla arriendo](#t_2_16)

[Tabla 2-17. Historial tarifa](#t_2_17)

[Tabla 2-18. Tabla proveedores](#t_2_18)

[Tabla 2-19. Tabla equipo](#t_2_19)

[Tabla 2-20. Tabla estado equipo](#t_2_20)

[Tabla 2-21. Tabla tipo equipo](#t_2_21)

[Tabla 3-1. Listado de programas](#t_3_1)

# ÍNDICE DE DIAGRAMAS

[Diagrama 1-1. DFA proceso venta](#d_1_1)

[Diagrama 1-2. DFA proceso control de equipos](#d_1_2)

[Diagrama 1-3. DFA proceso control de stock](#d_1_3)

[Diagrama 1-4. Diagrama de contexto general](#d_1_4)

[Diagrama 1-5. Diagrama de primer nivel](#d_1_5)

[Diagrama 3-1. Diagrama de menú](#d_3_1)

[Diagrama 3-2. Diagrama modular](#d_3_2)

[Diagrama 3-3. Diagrama de bloque login](#d_3_3)

[Diagrama 3-4. Diagrama de bloque menú principal](#d_3_4)

[Diagrama 3-5. Diagrama de bloque ventas](#d_3_5)

[Diagrama 3-6. Diagrama de bloque cuadrar caja](#d_3_6)

[Diagrama 3-7. Diseño de bloque de ingreso de factura](#d_3_7)

[Diagrama 3-8. Diagama de bloque mantenedor de trabajadores](#d_3_8)

[Diagrama 3-9. Diagrama de bloque arriendos](#d_3_9)

[Diagrama 3-10. Diagrama de bloque arrendar](#d_3_10)

# INTRODUCCIÓN

Desde mediados de la década de los 80, comenzó la masificación de las computadoras personales. Esto produjo precios más accesibles para la gente.

Llegando a 1995, las empresas comenzaron a brindar acceso a Internet vía red telefónica. Viendo esto como una buena oportunidad económica, algunos entusiastas emprendedores comenzaron a arrendar acceso a Internet por medio de cabinas de computadoras.

La familia Sepúlveda, buscando una forma de tener ingresos extras, se percató de que en la villa donde viven no hay acceso a Internet a un bajo costo. Las familias no pueden solventar el precio que las empresas proveedoras piden.

Con esto en mente, la familia Sepúlveda decidió implementar un cibercafé. Así nace, en la Villa Rukán, el cibercafé Interplay. El plan de los dueños consiste en proveer servicios de fotocopias, abarrotes artículos de oficina, Internet y videojuegos.

Tras un tiempo en el negocio, apareció la necesidad de utilizar un software para el manejo de las horas de arriendo de los computadores, más precisamente CabControl. Pero, este software no fue totalmente útil, ya que no cubría todas las necesidades del negocio.

Para este Trabajo de Título, se creará un Sistema de Administración de cibercafé, con procesos en el área de ventas, de control de stock, control de arriendo de computadoras y consolas de videojuego. La finalidad del sistema será facilitar a los dueños la tarea de administración.

# CAPÍTULO 1: ASPECTOS RELEVANTES DEL DISEÑO LÓGICO

**1. ASPECTOS RELEVANTES DEL DISEÑO LÓGICO**

## 1.1. DESCRIPCIÓN DE LA ORGANIZACIÓN

El cibercafé “Interplay” es una microempresa familiar, ubicada en Pichilemu Nº 110, Villa Rukán, Reñaca Alto, Viña del Mar. Surgió en el año 2008.

Esta pequeña empresa se dedica a proporcionar servicios de: arriendo de computadores con acceso a Internet, arriendo de PlayStation 2 (dentro del local), impresión de documentos, fotocopias y venta de artículos de escritorio y abarrotes.

Los dependientes del local están compuestos por don Jorge Sepúlveda, dueño del local, y doña Cecilia Zamora, esposa de don Jorge y administradora del local. Además, cuentan con la ayuda eventual del hijo del matrimonio, don Jorge Sepúlveda.

La idea de crear una microempresa surgió debido a que la familia Sepúlveda Zamora no contaba con el suficiente dinero para cubrir todos sus gastos: necesitan aumentar sus ingresos.

Es así que, en su población, vieron una buena oportunidad de generar recursos en arrendar servicios de Internet y juegos (de PlayStation 2, específicamente), y vender artículos de escritorio y abarrotes.

Para esto adquirieron 6 computadoras, de los cuales, cinco son para uso de los clientes; el otro hace la labor de servidor, el cual es utilizado por el encargado del negocio, conectados por un hub en una red LAN, y un PlayStation 2.

Después de un tiempo, en vista de la prosperidad del negocio, decidieron adquirir una multifuncional, la cual sirve como impresora y fotocopiadora. Esta pieza de hardware está conectada al computador servidor.

El cibercafé cuenta con un contador, externo al local, que se encarga del llenado y mantención de los libros contables: el libro mayor, menor, de ventas; además de asesorar a Interplay tributariamente.

## 1.2. DESCRIPCIÓN DE LA SITUACIÓN ACTUAL

Un cliente llega hasta las dependencias del local (en su mayoría vecinos del sector), camina hasta el mostrador y comunica al encargado de la caja qué es lo que desea (el encargado de la caja es algún miembro de la familia sostenedora).

El cliente puede escoger entre: arriendo de un computador, arriendo de la PlayStation2, impresión de archivos, fotocopia de documentos o comprar artículos de escritorio y/o abarrotes.

El encargado, entonces, le dice al cliente el precio de(los) servicio(s) y/o producto(s) solicitados. Pero, puede que el encargado no conozca el precio de algún ítem. En ese caso, tiene la posibilidad de consultar a la administradora del local. La señora Zamora habitualmente se encuentra en local, que es parte de su domicilio, durante los horarios de atención. Si ella no recordara el precio de algún producto, comunica al encargado un precio estimado del artículo en cuestión.

Hay algunos productos que tienen su precio rotulado con una calcomanía. Sin embargo, no hay ningún criterio sobre qué rotular ni con qué frecuencia.

Cuando el negocio recién comenzó, el local contaba con un listado con los precios de venta. Pero, después de un tiempo, ese listado desapareció. Según relatan los propios dueños del local: “Usar la lista tomaba mucho tiempo y era molesto. Terminamos no usándola y se perdió”.

Las fotocopias y las impresiones son trabajadas como productos comerciales. Cada hoja de fotocopia, o impresión, a blanco y negro, cuesta 50 pesos; mientras que la hoja a color, 100 pesos.

Para procesar la venta, el encargado de la caja busca un trozo de papel, de diferente índole y escribe en éste el subtotal para cada producto. El subtotal para un producto se obtiene multiplicando el precio de venta por la cantidad vendida.

El encargado, entonces, suma todos los subtotales, calcula el total y procede a comunicárselo al cliente. Todos los cálculos son realizados de manera manual: con papel y lápiz.

El cibercafé entrega boleta por la venta de productos, sean abarrotes (por ejemplo, una galleta o un paquete de papas fritas) u artículos de escritorio (como un CD, DVD o lápiz de pasta), y por el arriendo de PlayStation y computadores.

Si el total de la venta es menor o igual que ciento noventa pesos ($190), el cibercafé no entrega boleta. El local llevaba registro de estas ventas “menores” en un cuaderno. Sin embargo, después de un tiempo, no siguieron registrándolas.

El cibercafé sólo admite pagos en efectivo.

Una vez que haya recibido el dinero, el encargado llena una boleta, autorizada por el SII, y escribe en ella el total y la fecha actual. Luego, separa la boleta del talonario y se la entrega al cliente.

El cibercafé cuenta con seis computadores. Todos funcionan bajo el S.O. Windows XP. Cinco de éstos los pueden utilizar los clientes, siempre y cuando hayan pedido autorización al encargado. Los computadores están conectados mediante un Hub a una red LAN ETHERNET. Este hub está, también, conectado a un modem.

En cuanto al arriendo de un computador, el cliente puede escoger entre dos modalidades de tiempo: tiempo fijo y libre:

1. El tiempo fijo consiste en que el cliente paga el precio un cierto tiempo de arriendo: desde 10 minutos a varias horas. Una vez que el cliente paga por su arriendo, el encargado le asigna un computador por el tiempo transado.

El cliente tiene la posibilidad de ocupar el tiempo fijo de manera fraccionada en días distintos. Para ello, el cibercafé lleva un registro de los clientes con tiempo en reserva (sin utilizar) en un cuaderno.

Sin embargo, no es una práctica recurrente que los cliente vuelvan a usar el tiempo en reserva (ya sea porque no vuelven nunca más o porque se olvidan). El cibercafé dejó, entonces, de llevar registro de estos tiempos sin usar.

1. Mientras que en el tiempo libre, el usuario pide la utilización de un computador sin especificar el tiempo de uso. Luego, el encargado le asigna un computador. Aquí comienza a contar el tiempo de uso. Al finalizar el uso, se verifica el tiempo utilizado y se procede a pagar cuánto corresponda.

Para poder controlar que los tiempos asignados a los arriendos de los computadores sean cumplidos debidamente, el cibercafé hace uso de un programa de control de cabinas para cibercafés llamado CabControl.

Este programa está instalado en cada uno de los computadores del cibercafé, en los computadores de los clientes, y en el que utiliza el encargado.

La versión de CabControl usada es arcaica (data de un antes del 2008). Es la primera versión del software.

También, permite prender y apagar los equipos remotamente, contabilizar el tiempo de uso para cada usuario y mostrar por pantalla el precio de la hora (aunque no queda guardado).

Al programa se le asigna el valor de la hora de arriendo, la cual está establecida en cuatrocientos pesos ($400). Al momento de finalizar la sesión del cliente, CabControl muestra el costo de la sesión. Este se calcula multiplicando el tiempo utilizado (en minutos) por el valor de la hora dividido en 60 (minutos).

Además, es capaz de mostrar una tabla con información de los equipos clientes. Esta contiene: nombre del equipo, hora de inicio, hora de fin, tiempo fijado, precio, si el computador está ocupado o disponible y si el cliente usa tiempo fijo o libre.

El proceso para usar la PlayStation2 con la que cuenta el local es similar al de arriendo de un computador: sólo que el tiempo de uso debe ser contabilizado por el encargado de la caja. El precio por una hora de uso es de seiscientos pesos ($600).

Cuando el local queda desabastecido de algún producto, los dueños se dirigen, personalmente, al comercio mayorista para reponer sus repisas. Una vez se encuentran en allí (generalmente en Santiago), los dueños proceden visitar distintas tiendas, en busca de cual(es) les ofrece(n) el mejor precio para cada producto.

Cuando encuentra el local comercial más barato, compran la cantidad estimada conveniente de productos.

No existe ningún criterio o método para determinar a ciencia cierta cuántas existencias hay de un producto, ni cuántas comprar: sólo son estimadas de manera visual, a criterio de los dueños y/o de acuerdo a la demanda que tenga(n) los producto(s).

De vuelta en el local, los dueños reponen los productos que estén visiblemente en poca abundancia en el cibercafé. El resto es guardado en la casa de la familia, dentro de la misma construcción: inmediatamente al lado del local.

Si un cliente llega al local y trae consigo un producto comprado en el cibercafé, que estuviera en mal estado, puede pedir el cambio del producto. El cliente debe traer consigo la boleta asociada de la venta.

Para esto, el encargado de la caja realiza una inspección visual para constatar que lo que el cliente alega sea cierto. Si es así, el encargado le entrega al cliente un producto de igual características (si posee alguno en stock), ofrecerle otro producto o devolverle el valor del producto al cliente.

Esta opción queda a criterio del cliente (a menos que no haya un producto igual para reemplazar el defectuoso).

## 1.3. PROBLEMAS DETECTADOS

En base a las entrevistas con el usuario, y la observación propia, se han detectado las siguientes falencias:

* El programa de control de cabinas presenta fallas técnicas (direccionamiento IP, tiempo de respuesta, tiempo de uso), que entorpecen o imposibilitan trabajar cómodamente día a día.
* No hay cómo saber qué cliente tiene minutos a su favor. Los registros de tiempos en reserva no están actualizados (en cuaderno).
* No se pueden guardar los precios históricos de los productos
* No es posible saber cuántos productos existen en el local.
* El programa CabControl trabaja sobre una interfaz de usuario gráfica que no es agradable visualmente
* El programa CabControl tiene un tiempo de respuesta excesivo para su uso.
* En ocasiones, la interfaz de usuario de CabControl no responde a las acciones del usuario (específicamente, los botones de la interfaz), resultando en frustración y estrés para el cajero, además de demora en la atención al cliente.
* No existe ninguna manera de controlar rigurosamente el tiempo de uso de la PlayStation2
* No hay forma de saber cuándo quedan escasas existencias de un producto
* El cálculo del vuelto de una venta puede ser inexacto, ya que debe ser realizado por el encargado de la caja manualmente.
* No existe ninguna clasificación de los productos poseídos.
* La rotulación de los precios de los artículos es arbitraria y poco frecuente.
* El programa CabControl no es capaz de cerrar las aplicaciones de un equipo cliente cuando el tiempo de uso ha llegado a cero.
* El programa CabControl no posee perfiles de usuario, por lo que no tiene seguridad y cualquier persona malintencionada podría asignar tiempo a los computadores.
* CabControl no guarda bien el tiempo asignado a un equipo: a veces registra más; a veces, menos.
* Otras veces, CabControl desconfigura las direcciones IP de los computadores clientes y no pueden asignárseles un usuario ni tiempo de uso. Entonces, el encargado de la caja, o algún miembro de la familia Sepúlveda Zamora, debe re-configurar, manualmente, la dirección IP de cada equipo.

## 1.4. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

### 1.4.1. Objetivo principal

Desarrollar un sistema informático que permita administrar el arriendo de computadores y consolas de video juegos (PlayStation2), controlar las ventas de productos y apoyar el control de stock del cibercafé Interplay.

### 1.4.2. Objetivos específicos

Los objetivos específicos del sistema son los siguientes:

* Apoyar en el proceso de ventas de productos.
* Administrar el uso de los computadores y la PlayStation2 y conocer quién usa qué y por cuánto tiempo.
* Llevar un registro de los clientes.
* Emitir estadísticas acerca de las ventas.
* Producir informes que permitan a los dueños conocer cómo van las ventas.

### 1.4.3. Beneficios del sistema

El sistema permitirá:

* Mejorar los tiempos de respuesta.
* Conocer certeramente qué productos hay y cuántos.
* Saber con exactitud los precios de venta.
* Agilizar el proceso de ventas.
* Rescatar información de una venta, después de que ésta haya ocurrido.

### 1.4.4. Descripción general de la solución propuesta

Se desarrollará un sistema de administración comercial, que pueda resolver los problemas detectados y satisfaga las necesidades planteadas. Ayudará a la gestión del negocio, ya que será automatizado.

Se implementará una base de datos, donde se almacenarán los datos a utilizar. Dentro del cibercafé se utilizará una red LAN, con arquitectura cliente-servidor. Esto a futuro podrá ampliarse para ofrecer más computadores para los clientes.

En cuanto a la administración de arriendos de computadores y PlayStation2, el sistema pedirá recoger datos acerca de la persona que usará un equipo, la registrará o buscará en la base de datos y otorgará acceso al equipo en cuestión.

El procesamiento de ventas funcionará de la siguiente forma: el cliente pedirá el(los) producto(s) que desee y el cajero los registrará en el sistema, en la ventana de ventas. El sistema generará una venta, calculará el total y mostrará el vuelto que debe dar el cajero.

El sistema soportará el ingreso de facturas de compra de proveedores y los productos asociados a ésta, mediante la ventana de ingreso de facturas, solicitando a algún administrador que ingrese el código, nombre, cantidad y precio de los artículos. Una vez haya terminado de ingresar los datos, el sistema actualizará las existencias de cada producto involucrado.

Los productos podrán ser organizados en categorías. Inicialmente, el sistema no ofrecerá categorías por defecto. Eso quedará a criterio de los clientes.

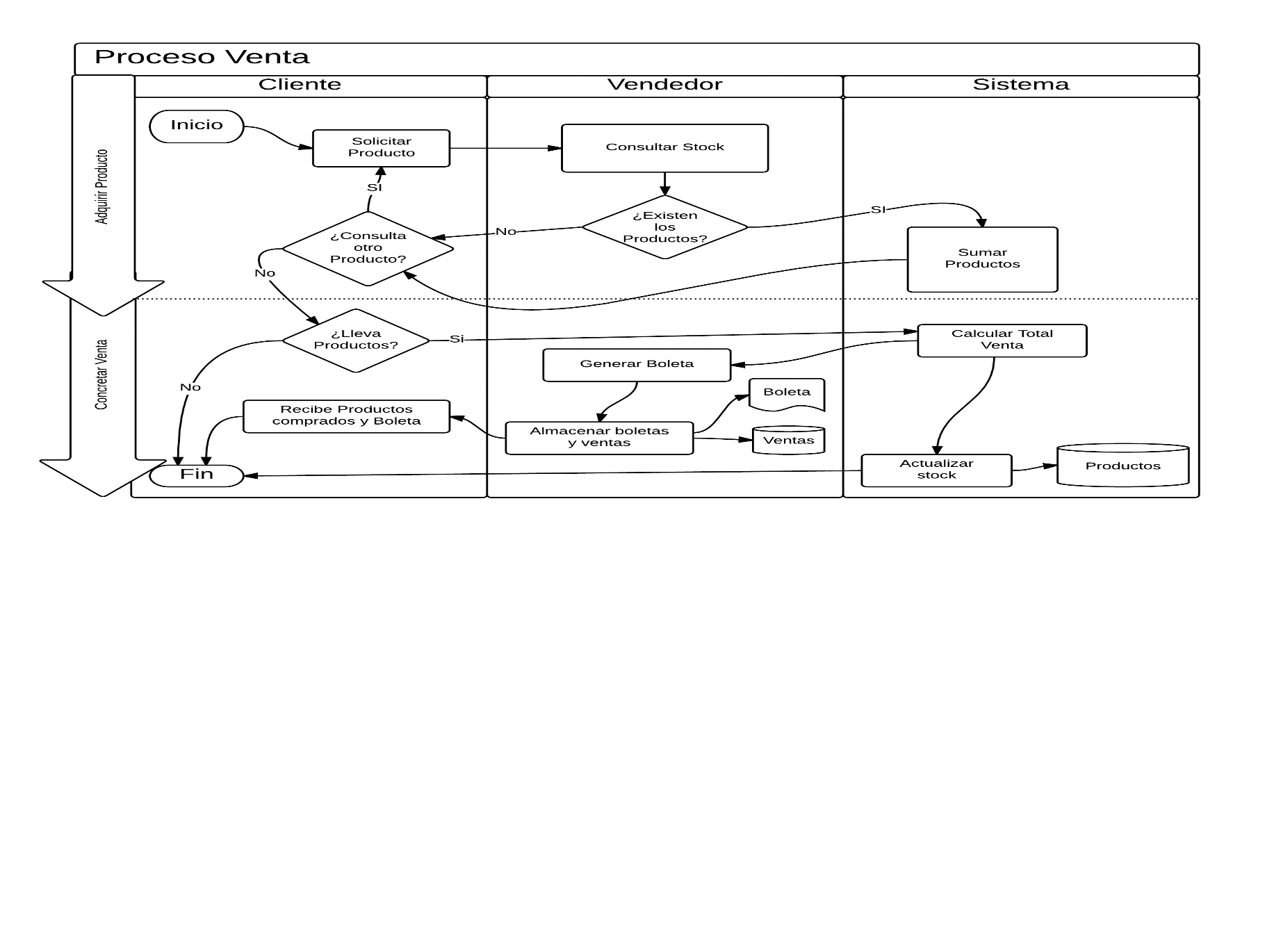
Con el fin de hacer seguro el sistema, se implementarán perfiles de usuario. Estos son: cajero (que puede realizar ventas y arriendos de computadores y PlayStation, y agregar usuarios) y administrador (que puede realizar ventas y arriendos de computadores y PlayStation, modificar usuarios, productos, facturas y precios).

Así, sólo los usuarios calificados podrán modificar datos de alta importancia como, por ejemplo, el valor de la tarifa de la PlayStation2.

### 1.4.5. Diagrama de flujo administrativo para el sistema propuesto

Para el sistema propuesto, se mostrarán los distintos diagramas de flujo administrativos, para los tres procesos. Los cuales son, Proceso Ventas, Proceso Control de Equipos, proceso control de stock.

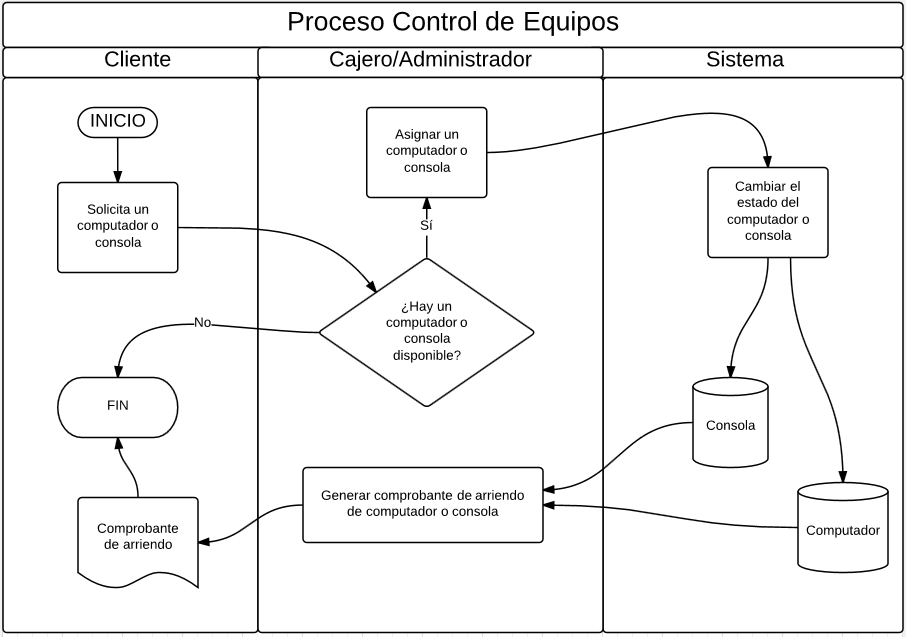
1. DFA Proceso ventas.



Fuente: Elaboración propia, basado en proceso de realizar una venta

Diagrama 1-1. DFA proceso venta

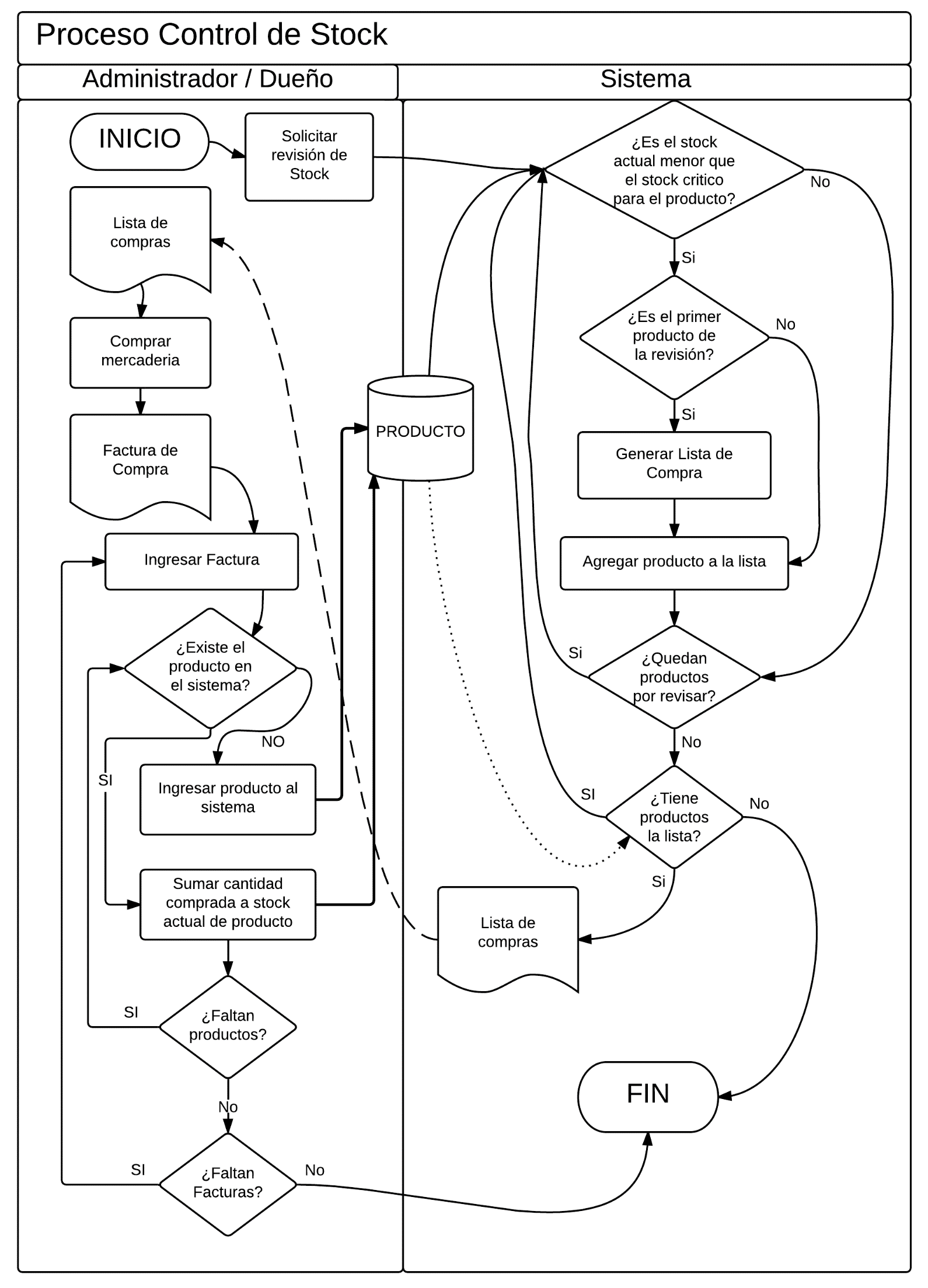
1. DFA Proceso control de equipos



Fuente: Elaboración Propia, basado en proceso para asignar un computador

Diagrama 1-2. DFA proceso control de equipos

1. DFA Proceso control de stock



Fuente: Elaboración Propia, basado en proceso de revisión de inventario

Diagrama 1-3. DFA proceso control de stock

### 1.4.6. Estructura funcional del sistema

Los procesos generales que efectuará el sistema serán:

1. Proceso de ventas
2. Proceso control de equipos
3. Proceso control de stock

### 1.4.7. Detalle de las funcionalidades por cada proceso

Las funciones específicas de cada proceso serian:

1. Proceso venta

Este proceso se encargará de todo lo relacionado con ventas y la caja del cibercafé. Tendrá cuatro funciones:

* **Venta:** Función que permite realizar una venta de productos o servicios a los clientes. Primero, antes de la venta, se deberá identificar si el cliente figura en la base de datos. De no estar, se le consultará al cliente si desea ser registrado en ésta. Después de esto, se procede a seleccionar los productos a vender desde una lista, la cual será mostrada por pantalla, digitando su código.

Luego, el vendedor deberá digitar la cantidad de artículos a vender de cada producto.

El programa permitirá borrar uno o más productos de la lista, en caso de ser necesario.

Al finalizar la venta, el sistema registrará la venta en la base de datos. El trabajador deberá escribir el precio en una boleta del talonario (autorizada por el SII), y entregarla al cliente. Estas boletas, la que se le entrega al cliente y la que genera el sistema, tendrán el mismo número de boleta.

* **Anular venta:** Cuando un cliente no esté conforme con una compra, se verificará el estado del producto y el número de la boleta de aquella venta. Se ingresará el número de boleta anulada al sistema más la descripción (motivo) de la devolución.
* **Consulta de producto:** Servirá al trabajador para hacer consulta sobre el producto ingresado, solicitando su código. Mostrará su precio, stock, categoría y descripción.
* **Cuadrar caja:** Al momento de apertura o cierre de caja, se utilizará esta función para ver los movimientos de caja, mostrando la diferencia de dinero, respecto a la apertura de la caja. Se deberá comparar con el dinero que hay en la caja.

1. Proceso control de equipos:

Este proceso poseerá funciones que muestren una lista de cada computador o videojuego que utilizan los clientes del cibercafé con la siguiente información: el nombre o identificación del equipo, el nombre de usuario del cliente que lo está usando, el estado actual del PC, la hora de inicio, el tiempo que lleva utilizado en esta sesión, el tiempo que queda para terminar la sesión, el tiempo de finalización y el precio que debe pagar. Esta lista se actualizará cada segundo mientras se estén utilizando los computadores. Tendrá cuatro funciones:

* **Mantenedora de equipos:** Permitirá agregar, modificar, eliminar y consultar equipos de la lista de equipos de los clientes. Se deberán ingresar los datos del equipo.
* **Informe de equipos:** Mostrará un registro histórico del uso de cada equipo, mostrando el tiempo utilizado por periodo de tiempo, por sesión, la última vez que fue utilizado, quien lo usó y en qué momento.
* **Asignar equipo a un cliente:** En el momento que un cliente pida utilizar un computador o un Videojuego, se le asignará uno que esté sin uso. El cliente deberá estar registrado en la base de datos, para poder utilizar un equipo. El tiempo de uso del equipo puede ser tiempo libre o fijo. Tiempo fijo significa que el cliente cancela de inmediato una cierta cantidad de tiempo. El sistema lo llevará a la función Ventas.
* **Cancelar equipo:** El cajero selecciona un equipo desde la ventana de aplicación. El equipo especificado será bloqueado y el tiempo de la sesión, detenido. A continuación, el cliente deberá pagar el tiempo utilizado, el sistema irá a la función ventas.

1. Proceso control de stock:

Este proceso se encargará del manejo del stock y lo reflejará en la base de datos.

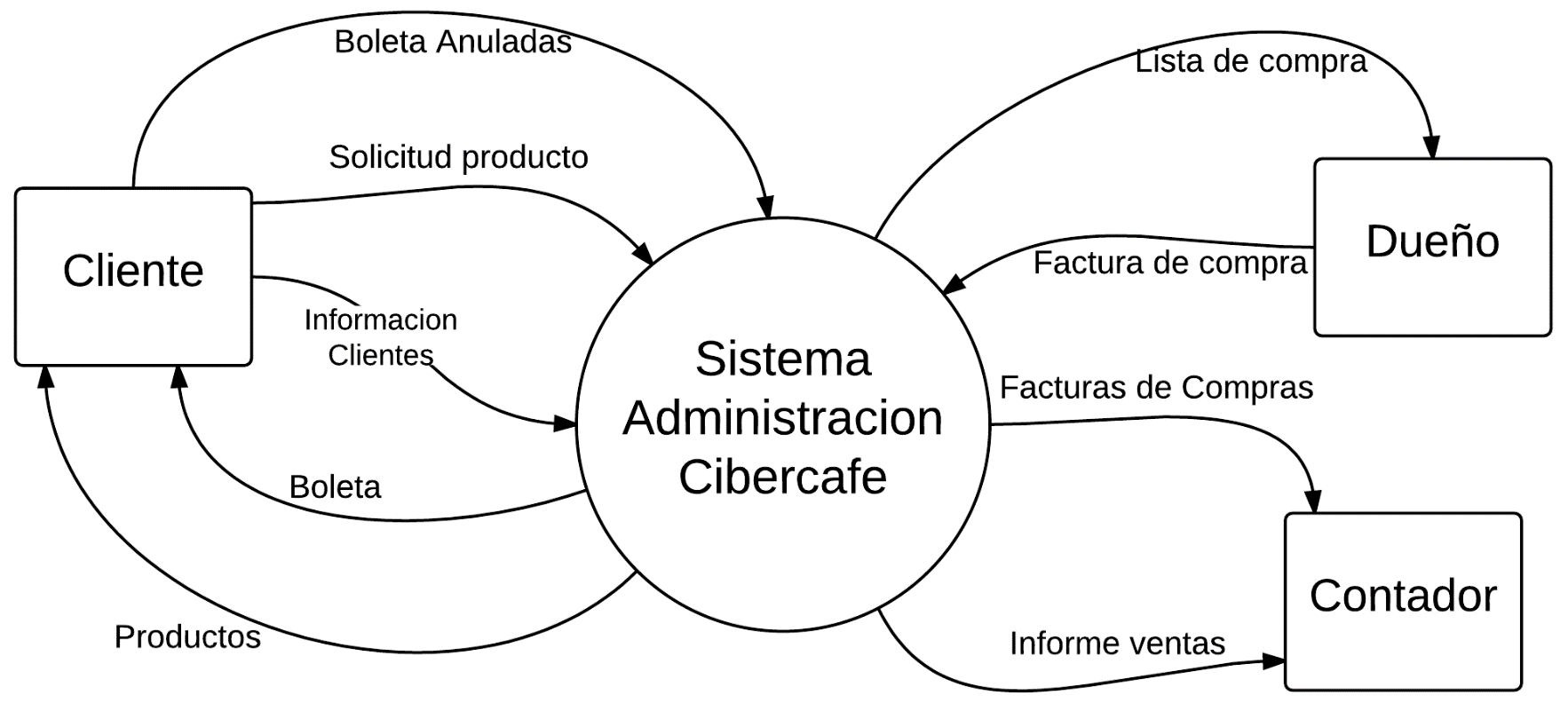
* **Mantenedor de productos:** Tendrá las opciones de Agregar, Modificar, Eliminar un producto. Para esto se ingresará su código o nombre del producto. Al agregar producto, se ingresará la cantidad, su categoría y stock crítico; si es que ya existe el producto se confirmará y se sumará al stock existente. Al modificar el producto, se podrá modificar el nombre y la cantidad del producto. Eliminar un producto, lo eliminará, lógicamente. Sólo un usuario de tipo administrador podrá eliminar un producto físicamente (acción irreversible).
* **Consultar Stock:** Tendrá dos opciones, la primera mostrará información un producto específico, y la segunda opción, información de todo el stock. Además, podrá generar informes y listado de los productos.
* **Mantenedor de Categorías:** Permitirá agregar, modificar, eliminar o consultar una categoría de productos. Deberá ingresar el código de categoría o el nombre de esta para administrarla.
* **Ajustar Stock:** Permitirá ajustar las existencias de productos, descontado o agregando productos del stock.
* **Mantenedor de facturas:** Permitirá ingresar, eliminar, actualizar y consultar una o más facturas de algún proveedor, traída al local por algún trabajador o por el dueño.
* **Asignar categoría a un producto:** Permitirá asociar uno o más productos a una categoría, escogida por el trabajador.
* **Generar informes:** El sistema generará informes sobre: historial de movimientos de inventario, producto más vendido, informe de stock crítico e informe de productos ordenado por stock.

1. Proceso control de usuarios:

* **Mantenedor de clientes** Se podrá agregar, modificar, eliminar y consultar sobre clientes. Estos serán registrados por su Rut, nombre, apellido, dirección, fono
* **Mantenedor de trabajadores:** Permitirá agregar, modificar, eliminar y consultar sobre trabajadores. Serán registrados por su Rut, nombre, apellido, dirección, fono y tipo de usuario (trabajador o administrador). Esto generará diferentes niveles de seguridad (perfiles de usuario).
* **Informe de clientes y trabajadores:** Mostrará información acerca del cliente. Mostrará cuanto tiempo ha pasado en el cibercafé, si posee tiempo a favor e información acerca de las compras del cliente.
* **Informe de clientes y trabajadores:** Mostrará información acerca de las acciones que haya realizado el usuario. Mostrará la misma información que para un cliente y visualizará qué acciones ha hecho en el cibercafé como, por ejemplo: ventas realizadas, computadores que haya asignado.

#### 1.4.7.1. Diagrama de contexto general

El siguiente diagrama, es el diagrama de contexto general. Se apreciará el sistema como un único proceso, el cual interactúa con los distintos agentes externos.

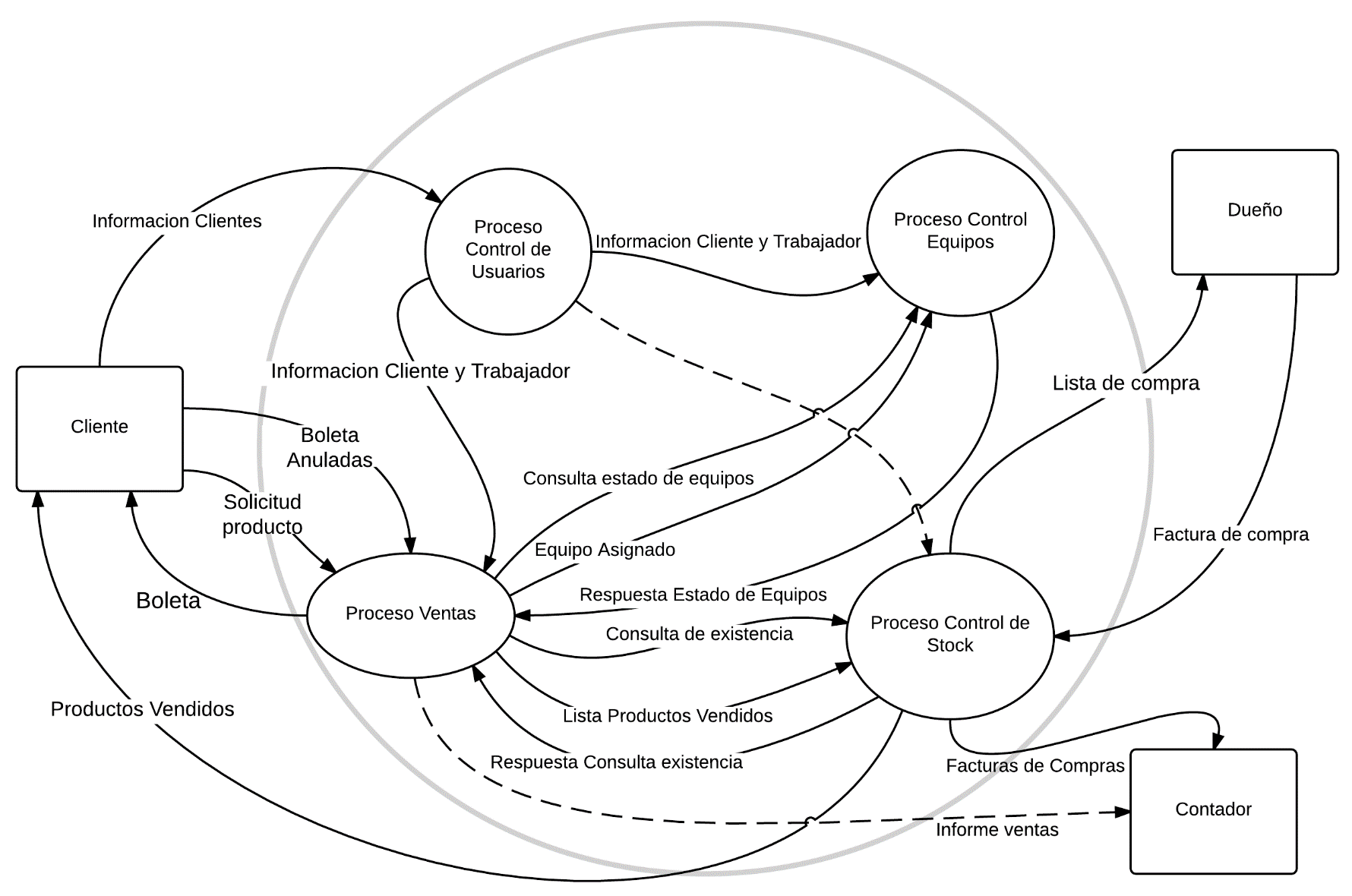


Fuente: Elaboración propia, basado en funcionamiento de la empresa

Diagrama 1-4. Diagrama de contexto general

#### 1.4.7.2. Diagrama de primer nivel

El sistema se divide en cuatro grandes procesos, los cuales se verán en el siguiente diagrama de primer nivel.



Fuente: Elaboración propia, basado en los procesos de la empresa

Diagrama 1-5. Diagrama de primer nivel

### 1.4.8. Manejo de información

El sistema de administración de cibercafé, utilizará diferentes datos, las cuales serían:

1. **Entradas:**

Son los datos que modifican el contenido de la base de datos, los cuales mayormente se ingresan vía teclado.

* + **Datos de clientes:** Rut, Nombre, Apellido, Dirección, Teléfono, Contraseña.
  + **Datos de usuarios:** Rut, Nombre, Apellido, Dirección, Teléfono, Contraseña, tipo de usuario.
  + **Datos de productos:** código de producto, nombre de producto, código de categoría, código de bodega
  + **Datos de ventas:** código de producto, cantidad de cada producto, precio de cada producto por cada venta, identificación de las boletas anuladas.
  + **Datos de arriendo:** fecha de la sesión, tipo de equipo que usará el cliente.
  + **Datos de ajuste de stock:** identificación del producto y cantidad a ajustar.
  + **Datos de computadores:** identificación de los computadores, dirección MAC, IP.
  + **Datos de consolas:** número de identificación de la consola.
  + **Datos de factura:** datos de la factura, el número de la factura, el Rut y nombre del proveedor.
  + **Datos de categoría:** nombre y descripción de la categoría.
  + **Datos de eetalle de factura:** datos de los productos que tiene cada factura.
  + **Datos de detalle de ventas:** datos de los productos que tiene cada venta, el código del producto y su cantidad.
  + **Datos de sesión**: datos de fecha y hora de inicio o término de la sesión de los trabajadores y de los clientes en los computadores.

1. **Salidas:**

Las salidas corresponden a las respuestas del sistema a la información requerida por el usuario. Estas serán mostradas por pantalla o impresas.

* **Listado de usuarios**: muestra información de los usuarios registrados en el sistema y el perfil de cada uno de ellos, mostrando sus datos personales, la cual pueden filtrarse en.
  + Listado de clientes
  + Listado de trabajadores
  + Listado de administradores
  + Informe individual de un cliente
  + Informe individual de un trabajador
  + Informe individual de un administrador
* **Listado de productos**: muestra una lista de los productos con su cantidad, código, nombre, unidades vendidas, y categoría a la que pertenecen y precio actual. Pueden verse de la misma forma las siguientes listas:
  + Listado de productos ordenados por stock
  + Listado de productos más vendidos
  + Listado de productos menos vendidos
  + Informe de un producto en específico.
* **Listado de equipos**: mostrará información de los equipos del cibercafé que utilizan los clientes. Pudiendo filtrarse en:
  + Listado de computadores
  + Listado de consolas
  + Informe de un equipo en particular.
* **Listado de precios:** mostrará el valor uno o varios de los productos o servicios de arriendo, según un filtro de fechas.
  + Informe de precios históricos de un producto
  + Listado de precio de todos los productos en un momento
  + Listado de precios históricos de arriendo
  + Informe de precio para un producto.
* **Listado de ventas:** mostrará información de las ventas, fecha, vendedor, cliente, número de boleta y total.
  + Listado de ventas con el total de cada boleta
  + Listado de ventas anuladas
  + Informe de una venta: encabezado más el detalle de cada producto.
  + Listado de ventas, ordenado por cliente
  + Listado de ventas mensuales
* **Listado de facturas:** mostrará información de las facturas ingresadas al sistema, numero de factura, Rut del proveedor, fecha. Se podrán listar:
  + Listado de facturas con total de la compra, ordenadas por fecha
  + Listado de facturas ordenado por proveedor.
  + Listado de facturas aún no pagadas.
* **Listado de locales:** mostrará información de los locales del negocio.
* **Informe de categorías:** muestra información de las categorías, y los productos asociados a ella.
* **Informe económico:** mostrará información estadística de las ganancias o pérdidas del cibercafé.
  + Informe de ganancias diarias
  + Informe de ganancias mensuales
  + Informe de pérdidas por venta anuladas

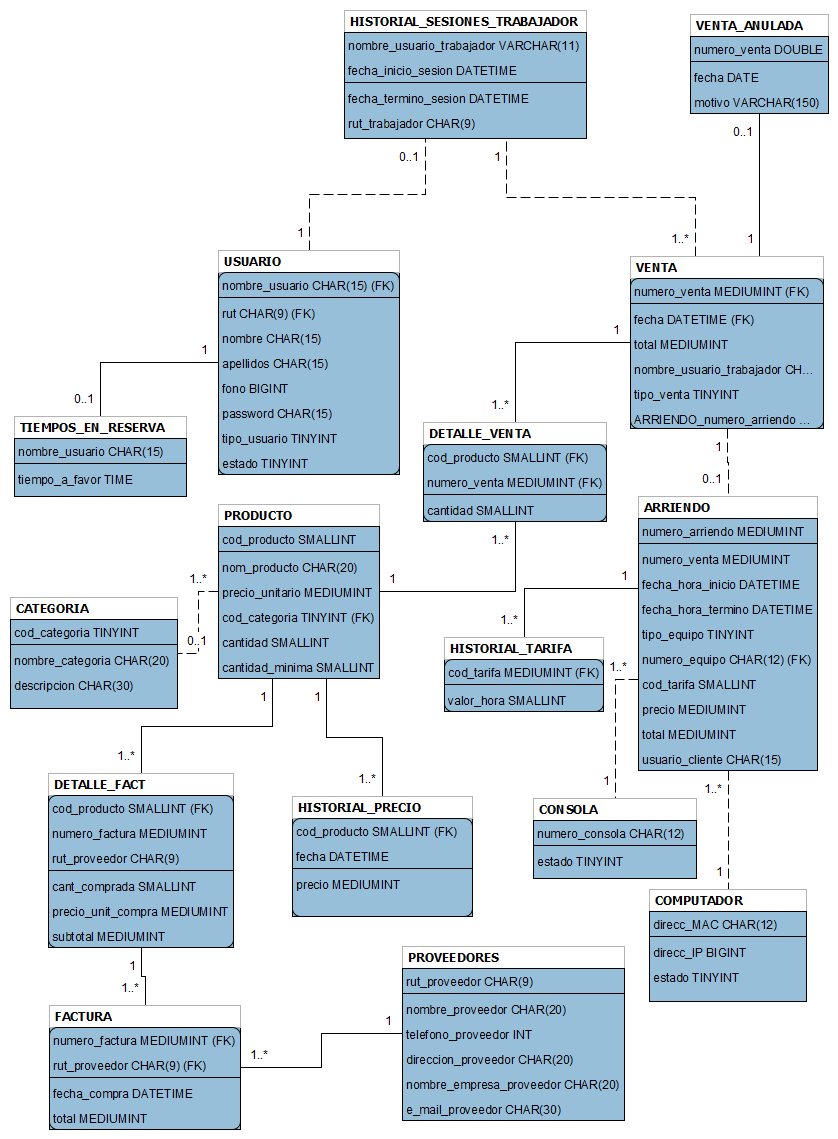
1. **Entidades de Información:**

Son las tablas que manejará la base de datos a implementar. Estas son:

* **Usuario**: Contendrá datos personales de cada usuario, trabajadores y clientes, la cual se diferenciará por el tipo de usuario, el que brindará niveles de seguridad.
* **Historial\_sesiones\_trabajador**: Tendrá datos históricos de los inicio de sesión de los trabajadores.
* **Arriendo**: Tendrá los datos de un arriendo, las que serían: fecha de la sesión, el código de la tarifa, el precio de la sesión, el número de venta y el número del arriendo, el cuales será correlativo.
* **Historial\_Tarifa**: Tendrá el historial del valor de la hora de los arriendo de equipos, en una fecha determinada y se le asignará un código de tarifa.
* **Computador**: Tendrá los datos de cada equipo cliente en la red, y su estado actual.
* **Consola**: Tendrá los datos de las consolas (videojuegos) del cibercafé.
* **Venta**: Tendrá el número de cada boleta, con el Rut del trabajador, del cliente, la fecha cuando fue emitida, su valor total neto, el local en el que se realizó, y que tipo de venta fue.
* **Venta\_Anulada**: Tendrá la fecha y motivo de las boletas anuladas, según el código de la boleta.
* **Detalle\_Venta**: Tendrá los códigos de cada artículo y su cantidad de cada boleta, según el número de venta.
* **Producto**: Tendrá el nombre, el código del producto, y el código de su categoría.
* **Categoria**: Tendrá el nombre, descripción y código de la categoría.
* **Historial\_precio**: Tendrá todos los precios antiguos de los productos.
* **Factura**: Tendrá datos de las facturas ingresadas
* **Detalle\_Fact**: Tendrá los detalles de las facturas ingresadas.
* **Proveedores:** Tendrá datos de los proveedores.

1. **Modelo de datos:**

Se presenta el modelo de datos relacional.

****

Fuente: Elaboración propia, basado en sistema de administración de cibercafé a realizar

Figura 1-1. Modelo de datos relacional “Sistema de Administración de cibercafé”

### 

### 1.4.9. Estructura de códigos

La codificación a utilizar para los códigos de los productos, código del local, códigos de las bodegas y códigos de las categorías será un código correlativo.

Para el código de la tarifa, será un código compuesto por la fecha de creación de la tarifa y el tipo de equipo en que se utiliza la tarifa. Por ejemplo, 2014-05-01#01, lo que significa que el 01-05-2014 se creó una tarifa para los computadores (#01).

El código de la categoría de los productos será definido por el usuario. Este código es de tipo numérico de máximo 3 dígitos, pudiendo éste ser desde 0 a 255.

El tipo\_usuario de la tabla usuario vendrá ya definido. Será un número de un dígito, el cual podrá ser: 1 para el administrador, 2 cajeros y 3 clientes.

El tipo\_venta de la tabla venta será un número de un dígito, el cual será 1 para la venta de productos, y 2 para un arriendo de equipo (consola o computador).

El tipo\_equipo de la tabla arriendo, será un número de dos dígitos, siendo éstos ya establecidos, los cuales son: 01 para arriendo de computador y 02 para arriendo de consola.

### 1.4.10. Condicionantes de diseño

El cibercafé tiene 6 computadores con el sistema operativo Windows XP, pero por la razón de que Microsoft ya no da soporte a este sistema operativo, se sugiere al cibercafé que migre a Windows 7.

* El sistema propuesto deberá operar bajo Windows 7 o posterior.
* Se utilizará el lenguaje de programación Visual BASIC en el IDE VisualStudio 2010.
* Se utilizará una base de datos MySQL.

# CAPÍTULO 2: MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS

**2. MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS**

En este capítulo se describirán los recursos computacionales de hardware y software a utilizar para desarrollar el sistema, y los equipos donde se utilizará el sistema. Se describirá también las entidades que se utilizarán en la base de datos para el sistema.

## 2.1. DESCRIPCIÓN DEL RECURSO COMPUTACIONAL

Se describe el recurso computacional el cual utilizará el sistema. Cada equipo del cibercafé tiene las mismas características de hardware y software.

### 2.1.1. Configuración del sistema

Los 6 computadores que posee el cibercafé Interplay tienen las mismas características de hardware y software.

* Procesador:

AMD Semprom 2600+ 32bits. 1.60 GHz

* Almacenamiento Primario:

704MB de RAM

* Almacenamiento Secundario:

80 GB

* Mecanismo de respaldo:

CD-RW

USB 2.0

* Pantallas:

1 pantalla LG L1718S

* Impresoras:

EPSON L355

* Red:

Puerto FastLan Ethernet

HUB para conectar los computadores vía cable Ethernet

### 2.1.2. Software utilizado

Se menciona el software a utilizar en el desarrollo del sistema.

1. Sistema operativo:

La aplicación será desarrollada para computadores con sistema operativo Windows 7. En primera instancia se iba a utilizar Windows XP, pero Microsoft dejó de brindar soporte técnico a Windows XP, por lo que se prefirió utilizar Windows 7.

Se utilizará en el cibercafé Interplay el sistema operativo Windows ya que los usuarios del cibercafé están familiarizados con éste, y muchas aplicaciones tienen compatibilidad con este sistema operativo. Sus características principales son:

* Compatibilidad: Windows 7 abstrae el proceso de instalación de más de 80.000 dispositivos de hardware interno y dispositivo de interfaz humana.
* Seguridad: Windows 7 incluye AppLocker que permite ejecutar aplicaciones que sólo están permitidos por el Administrador del Sistema.
* Ahorro de energía: Windows 7 reduce el consumo de energía hasta en un 30%.
* Cifrado de datos: Windows 7 incluye tecnología Bitlocker, que ofrece protección de datos de almacenamiento.

1. Lenguaje de programación

El sistema se desarrollará en el siguiente lenguaje:

Visual Basic: es un lenguaje de programación que se ha diseñado para facilitar el desarrollo de aplicaciones en un entorno gráfico para Windows.

Visual Basic es un lenguaje de programación visual, también llamado lenguaje de 4ª generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla.

Visual Basic es también un programa basado en objetos, utiliza objetos con propiedades y métodos, pero carece de los mecanismos de herencia y polimorfismo propios de los verdaderos lenguajes orientados a objetos.

Visual Studio es un entorno de desarrollo integrado, IDE. Soporta múltiples lenguajes de programación, como C++, C#, Visual Basic, .NET, F#, Java, Python, Ruby, PHP;

1. Sistema administrador de base de datos:

MySQL es un sistema de gestión de base de datos relacionales, multihilo y multiusuario.

MySQL es un software libre, desarrollado por MySQL AB y Oracle Corporation.

MySQL es usado por muchos sitios web como Facebook, YouTube, Wikipedia.

MySQL es utilizado en aplicaciones web, ligadas a PHP. MySQL es una base de datos muy rápida y es ideal para estos tipos de aplicaciones.

## 2.2. DESCRIPCIÓN DE ARCHIVOS

Se entrega un resumen de las tablas del sistema a realizar. Se describen los archivos utilizados por el sistema, especificando los campos de los registros y las claves primarias y foráneas de cada uno de ellos.

### 2.2.1. Listado de tablas

Tabla 2-1. Listado de tablas

|  |  |
| --- | --- |
| **Archivos** | **Descripción** |
| Usuario | Contendrá datos personales de cada usuario, trabajador y cliente. Podrá diferenciar tipos de usuario, lo que brindará niveles de seguridad. |
| Tiempos\_En\_Reserva | Tendrá el tiempo reservado de los clientes para utilizar en los computadores. |
| Historial\_Sesiones\_Trabajador | Tendrá datos históricos de los inicio de sesión de los trabajadores. |
| Arriendo | Tendrá los datos de un arriendo, ya sea de un computador o una PlayStation. |
| Historial\_Tarifa | Tendrá el historial del valor de la hora del arriendo de equipos (computador y PlayStation). Esto incluye la tarifa utilizada actualmente. |

Tabla 2-1. Listado de tablas (continuación)

|  |  |
| --- | --- |
| **Archivos** | **Descripción** |
| Computador | Tendrá los datos de cada equipo cliente en la red y su estado actual. |
| Consola | Tendrá los datos de las consolas (videojuegos) del cibercafé. |
| Venta | Contendrá datos de cada una de las ventas que realice el cibercafé. |
| Ventas\_Anuladas | Tendrá la fecha y motivo de las boletas anuladas, según el código de la boleta. |
| Detalle\_Venta | Detallará cada venta del cibercafé, descrita, a nivel general, en Venta. |
| Producto | Tendrá datos de los productos comercializados en Interplay. |
| Categoria | Contendrá datos de las categorías de productos. Estas deben ser creadas por el usuario. |
| Historial\_precio | Tendrá todos los precios históricos de los productos, más el precio actual. |
| Factura | Tendrá datos de las facturas ingresadas. |
| Detalle\_Fact | Detallará las facturas ingresadas en el sistema. |
| Proveedores | Tendrá datos de los proveedores. |

Fuente: Elaboración propia, basada en el modelo de datos

### 2.2.2. Codificaciones

Los códigos de los productos, código del local, códigos de las bodegas y códigos de las categorías a utilizar son números correlativos incrementales en uno.

El código de la tarifa, será un código compuesto por la fecha de creación de la tarifa y el tipo de equipo en que se utiliza la tarifa. Por ejemplo, 2014-05-01#01, lo que significa que el 01-05-2014 se creó una tarifa para los computadores (#01).

### 2.2.3. Tipos de datos

En los archivos se usarán los tipos de datos que existen en MySQL, porque éste será el motor de base de datos a utilizar en el sistema.

Los tipos de datos de MySQL se dividen en las siguientes categorías:

* Tipos numéricos
* Tipos de fechas y horas
* Tipos de cadenas de caracteres

Los tipos de datos a utilizar en el sistema serán:

#### 2.2.3.1. Tipos de datos numéricos

Tabla 2-2. Tipos de datos numéricos

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo | Bytes | Valor Mínimo | Valor Máximo |
|  |  | (con signo/sin signo) | (con signo/sin signo) |
| TINYINT | 1 | -128 | 127 |
|  |  | 0 | 255 |
| SMALLINT | 2 | -32768 | 32767 |
|  |  | 0 | 65535 |
| MEDIUMINT | 3 | -8388608 | 8388607 |
|  |  | 0 | 16777215 |
| INT | 4 | -2147483648 | 2147483647 |
|  |  | 0 | 4294967295 |
| BIGINT | 8 | -9223372036854775808 | 9223372036854775807 |
|  |  | 0 | 18446744073709551615 |

Fuente: Elaboración propia, basado en manual MySQL

#### 2.2.3.2. Tipos de datos fecha y hora:

Tabla 2-3. Tipos de datos fecha y hora

|  |  |  |  |
| --- | --- | --- | --- |
| DATE | 1000-01-01 | 9999-12-31 | 3 bytes |
| DATETIME | 1000-01-01 00:00:00 | 9999-12-31 29:59:59 | 8 bytes |
| TIME | -838:59:59 | 838:59:59 | 3 bytes |

Fuente: Elaboración propia, basado en manual MySQL

#### 2.2.3.3. Tipos de cadenas de caracteres

Tabla 2-4. Tipos de datos de cadena de caracteres

|  |  |
| --- | --- |
| CHAR (M) | M bytes.  Donde M es mayor o igual que 0 y menor o igual que 255.  Al ser guardado, el tipo CHAR concatena tantos espacios blancos como requiera para completar el largo (estricto) definido. |
| VARCHAR(M) | M= Largo\_del\_Registro + 1 bytes.  M es mayor o igual que 0 y menor o igual que 255. Por ejemplo, para la cadena 'abcd', el Largo\_del\_Registro es de 4 btyes y el espacio requerido para almacenarlo es de 5 bytes.  El tipo VARCHAR no necesita concatenar espacios en blanco al ser guardado (a diferencia del tipo CHAR). |

Fuente: Elaboración propia, basado en manual MySQL

### 2.2.4. Descripción de archivos

A continuación se describe cada archivo utilizado por el sistema a implementar.

#### 2.2.4.1. Usuario

Tabla 2-5. Tabla usuario

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Datos de la persona que utiliza el (ingresa al) sistema. | | |
| **CLAVE PRIMARIA** | Rut | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| Rut | CHAR | 9 | Rut del usuario, sin puntos ni guiones. Por ejemplo: 184576589 |
| Nombre | CHAR | 30 | Nombre del usuario |
| Apellidos | CHAR | 30 | Apellido del usuario |
| Fono | VARCHAR | 20 | Teléfono del usuario |
| Password | CHAR | 15 | Contraseña del usuario encriptada |
| tipo\_usuario | TINYINT | 1 | Tipo de usuario. 1: Administrador, 2: Cajero |
| Estado | TINYINT | 1 | Situación del usuario. 0: Activo, 1: Eliminado |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.2. Tiempos\_en\_reserva

Tabla 2-6. Tabla tiempos en reserva

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Llevará registro de los tiempos a favor (en reserva) que tengan los clientes del cibercafé. | | |
| **CLAVE PRIMARIA** | nombre\_usuario | | |
| **CLAVE FORANEA** |  | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| nombre\_usuario | CHAR | 15 | Cuenta del cliente |
| tiempo\_a\_favor | TIME |  | Tiempo para uso prepago del cliente |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.3. Historial\_sesiones\_trabajador

Tabla 2-7. Tabla historial sesiones trabajador

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DESCRIPCIÓN** | Tendrá datos históricos de los inicio de sesión de los trabajadores. | | | |
| **CLAVE PRIMARIA** | rut\_trabajador | | | |
| **CLAVE FORANEA** | rut\_trabajador (USUARIO.rut) | | | |
| **DESCRIPCIÓN DE REGISTRO** | | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** | |
| rut\_trabajador | CHAR | 9 | | RUT del trabajador que inicia sus funciones. No incluye puntos ni guion. Ejemplo: 184576589. |
| fecha\_inicio\_sesion | DATETIME |  | | Fecha y hora del inicio de sesión |
| fecha\_termino\_sesion | DATETIME |  | | Fecha y hora del termino de sesión |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.4. Venta

Tabla 2-8. Tabla venta

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Contendrá datos de cada una de las ventas (a nivel general). | | |
| **CLAVE PRIMARIA** | numero\_venta | | |
| **CLAVE FORANEA** | rut\_trabajador (USUARIO.rut) | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| numero\_venta | MEDIUMINT | 9 | Numero correlativo que identifica la venta |
| Fecha | DATETIME |  | Fecha cuando se realizó la venta |
| Total | MEDIUMINT | 9 | Total de la venta |
| rut\_trabajador | CHAR | 9 | RUT (Ej. 12345678K) del cajero involucrado en la venta. |
| tipo\_venta | TINYINT | 1 | Tipo de venta. 1: venta de productos, 2: arriendo de equipos, 3: venta de artículos de escritorio o fotocopias. |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.5. Venta\_anulada

Tabla 2-9. Tabla venta anulada

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Contendrá todas las ventas que hayan sido anuladas, explicando la razón de esta decisión. | | |
| **CLAVE PRIMARIA** | numero\_venta | | |
| **CLAVE FORANEA** | numero\_venta (VENTA.numero\_venta) | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| numero\_venta | MEDIUMINT | 9 | Número de la venta anulada, registrado en la tabla VENTA. |
| Fecha | DATETIME |  | Fecha de la anulación de la venta |
| Motivo | VARCHAR | 200 | Detalla porqué fue anulada la venta. |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.6. Detalle\_venta

Tabla 2-10. Tabla detalle venta

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Tendrá los código de cada artículo y su cantidad de cada boleta | | |
| **CLAVE PRIMARIA** | cod\_producto + numero\_venta | | |
| **CLAVE FORANEA** | cod\_producto(PRODUCTO. cod\_producto)  numero\_venta(VENTA.numero\_venta) | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| cod\_producto | SMALLINT | 5 | Referencia al código del producto en tabla PRODUCTO. |
| numero\_venta | MEDIUMINT | 9 | Referencia al número de la venta en la tabla VENTA. |
| cantidad | SMALLINT | 6 | Cantidad del producto vendido |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.7. Producto

Tabla 2-11. Tabla producto

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Detalles de cada producto | | |
| **CLAVE PRIMARIA** | cod\_producto | | |
| **CLAVE FORANEA** | cod\_categoria(CATEGORIA.cod\_categoria) | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| cod\_producto | SMALLINT | 5 | Número identificador del producto |
| nom\_producto | CHAR | 20 | Nombre del producto |
| precio\_unitario | MEDIUMINT | 9 | Precio del producto en pesos ($) |
| cod\_categoria | TINYINT | 3 | Referencia al código de la categoría |
| cantidad | SMALLINT | 6 | Cantidad actual del producto |
| cantidad\_minima | SMALLINT | 6 | Cantidad critica del producto |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.8. Categoría

Tabla 2-12. Tabla categoría

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Información de las categorías de los productos. | | |
| **CLAVE PRIMARIA** | cod\_categoria | | |
| **CLAVE FORANEA** |  | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| cod\_categoria | TINYINT | 3 | Identificador de la categoría |
| nombre\_categoria | CHAR | 20 | Nombre de la categoría |
| descripcion | VARCHAR | 200 | Descripción de la categoría (responde a: ¿Qué contiene?) |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.9. Detalle\_fact

Tabla 2-13. Tabla detalle\_fact

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Tendrá los detalles de las facturas ingresadas | | |
| **CLAVE PRIMARIA** | cod\_producto + numero\_factura + rut\_proveedor | | |
| **CLAVE FORANEA** | cod\_producto(PRODUCTO.cod\_producto)  numero\_factura (FACTURA. numero\_factura)  rut.proveedor(FACTURA. rut\_proveedor) | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| cod\_producto | SMALLINT | 5 | Hace referencia al código del producto |
| numero\_factura | MEDIUMINT | 9 | Hace referencia al número de factura |
| rut\_proveedor | CHAR | 9 | Rut del proveedor de la factura |
| cant\_comprada | SMALLINT | 6 | Cantidad del producto comprada |
| precio\_unit\_compra | MEDIUMINT | 9 | Precio unitario del producto |
| Subtotal | MEDIUMINT | 9 | Precio total de los productos comprados |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.10. Factura

Tabla 2-14. Tabla factura

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Tendrá datos de las facturas ingresada por concepto de comprar a proveedores. | | |
| **CLAVE PRIMARIA** | numero\_factura + rut\_proveedor | | |
| **CLAVE FORANEA** | rut\_proveedor(PROVEEDORES.rut\_proveedor) | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| numero\_factura | MEDIUMINT | 9 | Identificador de la factura |
| rut\_proveedor | CHAR | 9 | Rut del proveedor de la factura |
| fecha\_compra | DATETIME |  | Fecha de la factura |
| Total | MEDIUMINT | 9 | Precio total de la factura |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.11. Historial\_precio

Tabla 2-15. Tabla historial precio

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Tendrá todos los precios de los productos: el último ingresado y los antiguos. | | |
| **CLAVE PRIMARIA** | cod\_producto + fecha | | |
| **CLAVE FORANEA** | cod\_producto(PRODUCTO.cod\_producto) | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| cod\_producto | SMALLINT | 5 | Hace referencia al código del producto |
| Fecha | DATETIME |  | Fecha de la creación del precio |
| Precio | MEDIUMINT | 9 | Precio del producto |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.12. Arriendo

Tabla 2-16. Tabla arriendo

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Tendrá los datos de un arriendo, ya sea de computadores o de consolas. | | |
| **CLAVE PRIMARIA** | numero\_arriendo + numero\_venta | | |
| **CLAVE FORANEA** | cod\_tarifa (HISTORIAL\_ARRIENDO.cod\_tarifa)  numero\_venta (VENTA.numero\_venta)  usuario\_cliente(CLIENTE.nombre\_usuario) | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| numero\_arriendo | MEDIUMINT | 9 | Número correlativo identifica el arriendo |
| numero\_venta | MEDIUMINT | 9 | Hace referencia al número de venta |

Tabla 2-16. Tabla arriendo (continuación)

|  |  |  |  |
| --- | --- | --- | --- |
| fecha\_hora\_inicio | TIMESTAMP |  | Fecha y hora de inicio de arriendo |
| fecha\_hora\_fin | TIMESTAMP |  | Fecha y hora de termino del arriendo |
| id\_equipo | TINYINT | 4 | Identificador que apunta a la tabla EQUIPO |
| cod\_tarifa | SMALLINT | 6 | Hace referencia al código de la tarifa |
| Precio | MEDIUMINT | 9 | Costo de la tarifa |
| total\_precio\_arriendo | MEDIUMINT | 9 | Costo total del arriendo |
| usuario\_cliente | CHAR | 15 | Nombre de usuario del cliente |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.13. Historial\_tarifa

Tabla 2-17. Historial tarifa

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Tendrá el historial del valor de la hora | | |
| **CLAVE PRIMARIA** | cod\_tarifa | | |
| **CLAVE FORANEA** |  | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| cod\_tarifa | SMALLINT | 6 | Numero correlativo que identifica las distintas tarifas. |
| Fecha | TIMESTAMP |  | Fecha en la que se instauró una tarifa. |
| valor\_hora | SMALLINT | 6 | Valor en pesos ($) de una hora de arriendo de un equipo |
| id\_tipo\_equipo | TINYINT | 4 | Numero correlativo que apunta a la tabla TIPO\_EQUIPO. |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.14. Proveedores

Tabla 2-18. Tabla proveedores

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Tendrá los datos de los proveedores. | | |
| **CLAVE PRIMARIA** | rut\_proveedor | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| rut\_proveedor | CHAR | 9 | Rut del proveedor, sin puntos ni guion. Por ejemplo: 184576589 |
| nombre\_proveedor | CHAR | 30 | Nombre de encargado de proveedor. |
| nombre\_empresa\_proveedor | VARCHAR | 20 | Nombre de la empresa proveedora. |
| telefono\_proveedor | CHAR | 20 | Teléfono del proveedor. |
| direccion\_proveedor | CHAR | 20 | Dirección del proveedor. |
| e\_mail\_proveedor | CHAR | 30 | Dirección de email del proveedor. |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.15. Equipo

Tabla 2-19. Tabla equipo

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Contiene datos de todos los equipos con los que cuenta el cibercafé. | | |
| **CLAVE PRIMARIA** | rut\_proveedor | | |
| **CLAVE FORANEA** |  | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| id\_equipo | TINYINT | 4 | Número correlativo identificador de cada equipo |
| tipo\_equipo | INT | 4 | Número que apunta a la tabla TIPO\_EQUIPO y categoriza los equipos. |
| Descripción | VARCHAR | 45 | Contiene características técnicas del equipo en cuestión. |
| Estado | INT | 1 | Apunta a la tabla ESTADO\_EQUIPO e indica la disponibilidad del equipo. |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.16. Estado\_equipo

Tabla 2-20. Tabla estado equipo

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Contiene datos sobre los posibles estados de disponibilidad en los que puede estar un equipo del cibercafé. | | |
| **CLAVE PRIMARIA** | id\_estado | | |
| **CLAVE FORANEA** | id\_estado (EQUIPO.estado) | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| id\_estado | INT | 1 | Numero correlativo identificador de cada estado |
| nombre\_estado | VARCHAR | 45 | A qué estado corresponde (P.ej: Disponible) |

Fuente: Elaboración propia, basado en modelo de datos

#### 2.2.4.17. Tipo\_equipo

Tabla 2-21. Tabla tipo equipo

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN** | Contiene datos de las categorias de equipos disponibles | | |
| **CLAVE PRIMARIA** | id\_tipo\_equipo | | |
| **CLAVE FORANEA** | id\_tipo\_equipo (EQUIPO.tipo\_equipo) | | |
| **DESCRIPCIÓN DE REGISTRO** | | | |
| **NOMBRE** | **TIPO** | **LARGO** | **DESCRIPCION** |
| id\_tipo\_equipo | TINYINT | 4 | Número correlativo identificador de cada tipo (categoría) de equipo. |
| Descripción | VARCHAR | 150 | Descripción detallada del tipo de equipo, incluyendo características técnicas. |

Fuente: Elaboración propia, basado en modelo de datos

# CAPÍTULO 3: DESCRIPCIÓN DE PROGRAMAS

**3. DESCRIPCIÓN DE PROGRAMAS**

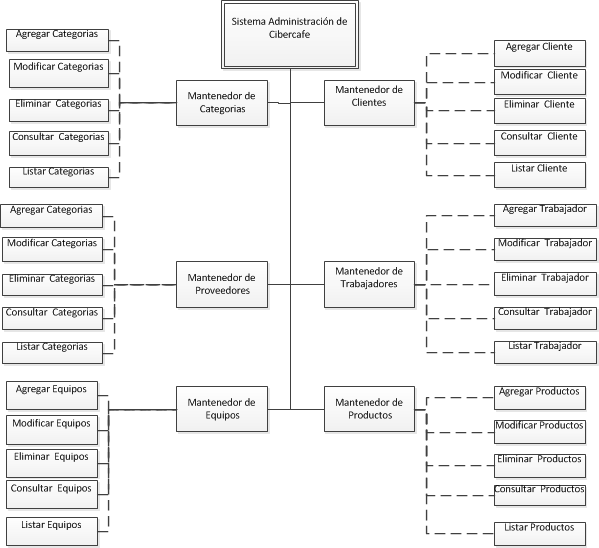
A continuación, se mostrará el diagrama de menués, el diagrama modular y la descripción de programas.

## 3.1. DIAGRAMA DE MENU

Fuente: Elaboración Propia, basado en los menues que utilizará el sistema a implementar

Diagrama 3-1. Diagrama de menú

## 3.2. DIAGRAMA MODULAR



Fuente: Elaboración Propia, basado en los modulos del sistema a implementar

Diagrama 3-2. Diagrama modular

## 3.3. LISTADO DE PROGRAMAS

Tabla 3-1. Listado de programas

|  |  |  |
| --- | --- | --- |
|  | **NOMBRE** | **DESCRIPCIÓN** |
| 1 | Login | Permite el ingreso al sistema, requiriendo nombre de usuario y contraseña del trabajador. |
| 2 | Menú Principal | Permite la ejecución de otros programas, donde el usuario interactúa con el Sistema. |
| 3 | Equipos en Arriendo | Muestra la lista de equipos, para asignar un arriendo o cancelar el arriendo |

Tabla 3-1. Listado de programas (continuación)

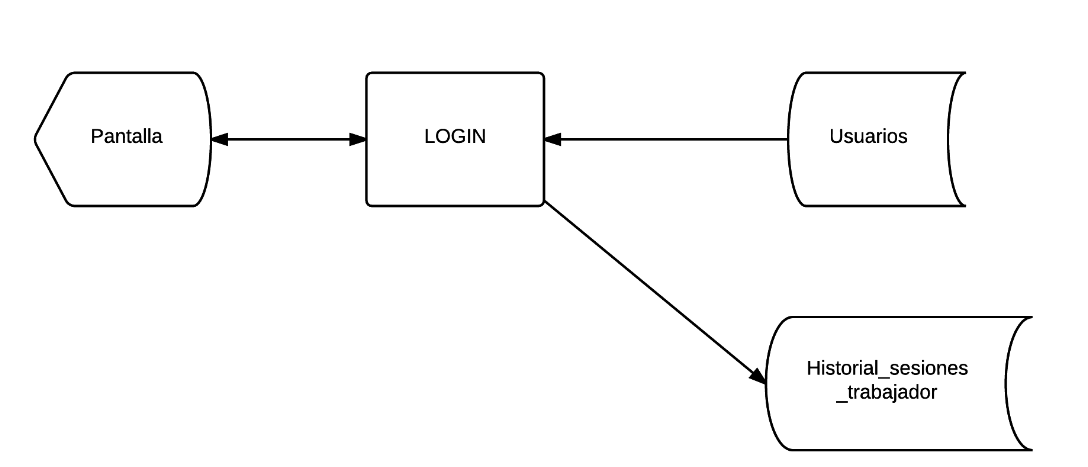
|  |  |  |
| --- | --- | --- |
|  | **NOMBRE** | **DESCRIPCIÓN** |
| 4 | Venta de Productos | Permite el ingreso de Productos a la lista de compra |
| 5 | Anular Venta | Permite el ingreso de número de boleta a anular y motivo por la anulación, se muestra listado de productos de la boleta. |
| 6 | Apertura Caja | Permite el ingreso cantidad inicial de dinero de caja |
| 7 | Ajustar Stock | Permite el ingreso de código de producto, y cantidad a descontar |
| 8 | Asignar Categoría a Producto | Se selecciona el producto y la categoría que se le asignará al producto |
| 9 | Consultar Stock | Mostrará información del producto ingresado |
| 10 | Cuadrar Caja | Muestra información para proceder el arqueo de caja |
| 11 | Ingreso Factura | Permite el ingresar una factura al sistema, con su listado de productos para actualizar el stock |
| 12 | Mantenedor Proveedores | Mostrará listado de proveedores y opciones de agregar, modificar y eliminar proveedores |
| 13 | Mantenedor de Categorías | Mostrará listado de Categorías y opciones de agregar, modificar y eliminar Categorías |
| 14 | Mantenedor de Clientes | Mostrará listado de Clientes y opciones de agregar, modificar y eliminar Clientes |
| 15 | Mantenedor Productos | Mostrará listado de Productos y opciones de agregar, modificar y eliminar Productos |
| 16 | Mantenedor Trabajadores | Mostrará listado de Trabajadores y opciones de agregar, modificar y eliminar Trabajadores |
| 17 | Mantenedor Equipos | Mostrará listado de Equipos y opciones de agregar, modificar y eliminar Equipos |
| 18 | Informe de Productos | Mostrará listado de Productos y opciones de imprimir o exportar a formato Excel PDF o Word |
| 19 | Informe de Stock Critico | Mostrará listado de Productos con falta de stock. |

Fuente: Elaboración propia, basado en procesos del sistema

## 3.4. DISEÑO DE PROGRAMAS

### 3.4.1. Login

* Nombre de programa: Login (Inicio de sesión).
* Objetivo: Restringir el acceso al sistema sólo a personas autorizadas.
* Diagrama de bloque:



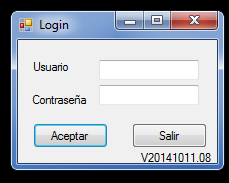
Fuente: Elaboración propia, basado en funcionamiento de login

Diagrama 3-3. Diagrama de bloque login

* Reglas de proceso:

1. Si el usuario presiona el botón “Salir”, se cierra la ventana de inicio de sesión y termina la ejecución del programa.
2. Si el usuario hace clic en “Aceptar”, se validará localmente los datos ingresados (RUT y contraseña), mediante una función booleana, donde:
   1. Si la casilla de rut está vacía, se manda un mensaje de error indicando ésto y la función retorna falso.
   2. Si no, si el rut no es válido, el sistema mandará un mensaje de error y la función retorna falso.
   3. O si no, si la casilla de contraseña está vacía, se manda un mensaje de error y la función se vuelve falsa.
   4. O si no, la función retorna verdadero (Los datos son válidos “localmente”).
   5. Si la función anterior retorna verdadero,
   6. El sistema busca en la base de datos, en la tabla USUARIO, el rut ingresado (Búsqueda por la clave primaria *rut*).
3. Si la contraseña y Rut coinciden con algún registro de la BD, se cierra la interfaz de login y se muestra la del menú principal. Además, se registra el inicio de sesión del trabajador en la BD. Más específicamente, en la tabla HISTORIAL\_SESIONES\_TRABAJADOR, creando una tupla e incializando los campos *rut* y *fecha\_inicio\_sesion (*el campo *fecha\_termino\_sesion* queda marcado como nulo).
4. si no, el sistema manda un mensaje de error, indicando un error de ingreso en los datos.

* Diseño de Pantalla:

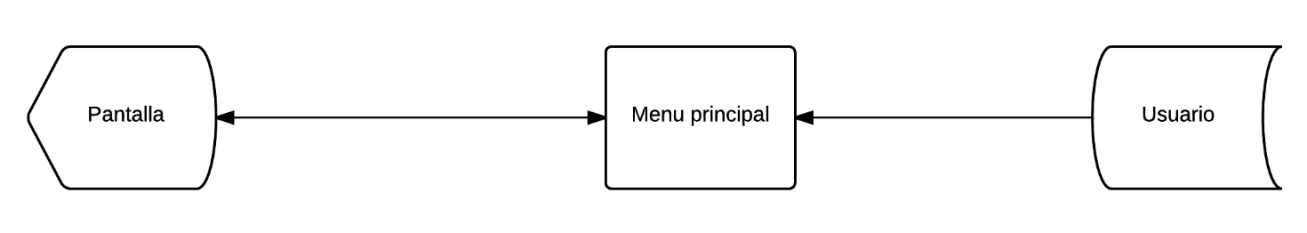


Fuente: Elaboración propia, basado en diseño de demostración 1

Figura 3-1. Diseño de pantalla login

### 3.4.2. Menú principal

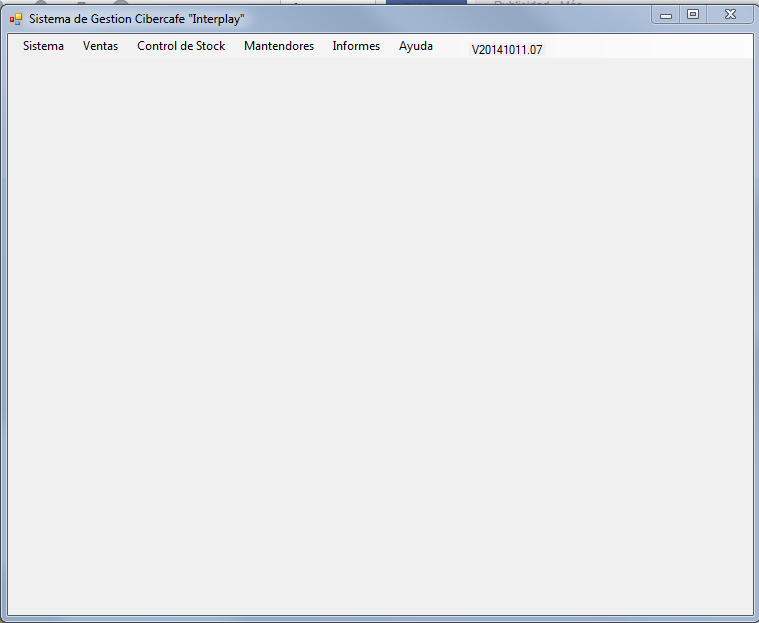
* Nombre del programa: Menú principal
* Objetivo del programa: Proveer una interfaz que permita al usuario utilizar todas las funcionalidades del sistema, de acuerdo a su perfil.
* Diagrama de bloque



Fuente: Elaboración propia, basado en funcionamiento del menú

Diagrama 3-4. Diagrama de bloque menú principal

* Reglas de proceso
  1. Al cargarse el menú, el sistema verificará el tipo de usuario (este dato está almacenado en una variable, el cual es obtenido, a su vez, de la tabla USUARIO).
  2. Si el usuario es un trabajador, el menú mostrará como disponibles las siguientes funcionalidades:
     1. Sistema
     2. Ventas
     3. Control de Stock
     4. Mantenedores
     5. Informes
     6. Ayuda
* Diseño de Pantalla

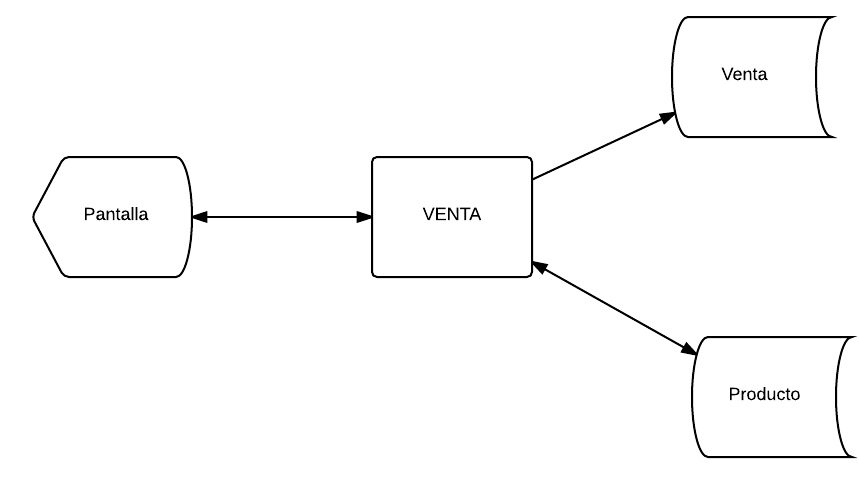


Fuente: Elaboración propia, basado en diseño de demostración

Figura 3-2. Diseño de pantalla menú principal

### 3.4.3. Venta de productos

* Nombre del programa: Ventas
* Objetivo: Concretar ventas de artículos y servicios ofrecidos por el cibercafé (arriendos, fotocopias, impresiones, abarrotes, artículos de escritorio).
* Diagrama de bloque:



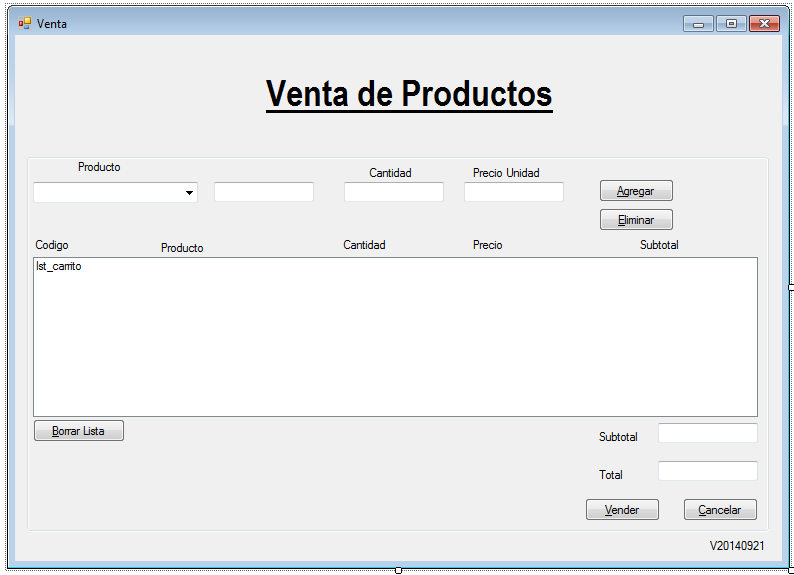
Fuente: Elaboración Propia, basado en funcionamiento de proceso de arriendo

Diagrama 3-5. Diagrama de bloque ventas

* Reglas de proceso

1. Al cargarse el programa Ventas, el sistema consultará, en la base de datos asociada, en la tabla PRODUCTOS por todos los productos disponibles (con stock mayor o igual que uno) para listarlos en las lista desplegable de búsqueda de productos, que permiten filtrar mediante el nombre y el código interno.
2. El programa, también, cargará el Rut del trabajador que ha iniciado sesión y lo mostrará en la casilla “Rut de trabajador”.
3. El cajero/trabajador puede agregar producto (o servicios) a la venta mediante el botón “Agregar”. Éste se cerciorará que el producto seleccionado tenga el stock suficiente para vender al cliente lo que este ha solicitado. De otro modo, el sistema mandará un mensaje de error, indicando esta situación.
4. A medida se agreguen productos a la venta, el sistema calculará el total de la venta y lo mostrará en la casilla “Total”.
5. El trabajador puede eliminar la lista de producto propuestos a ser vendidos, mediante el botón “Borrar lista”.
6. Si se hace clic en el botón “Salir”, la venta se cancelará, se perderá la lista de productos a ser vendidos.
7. Cuando el cajero hace clic en el botón “Vender”, el sistema:
   1. Descuenta de las existencias de cada artículo, la misma cantidad que son vendidos en la transacción actual o,
   2. Asigna un equipo al cliente, dependiendo de que haya solicitado.
8. En ambos casos, se genera una boleta de uso interno, no válida en el SII, y termina el programa.

* Diseño de Pantalla

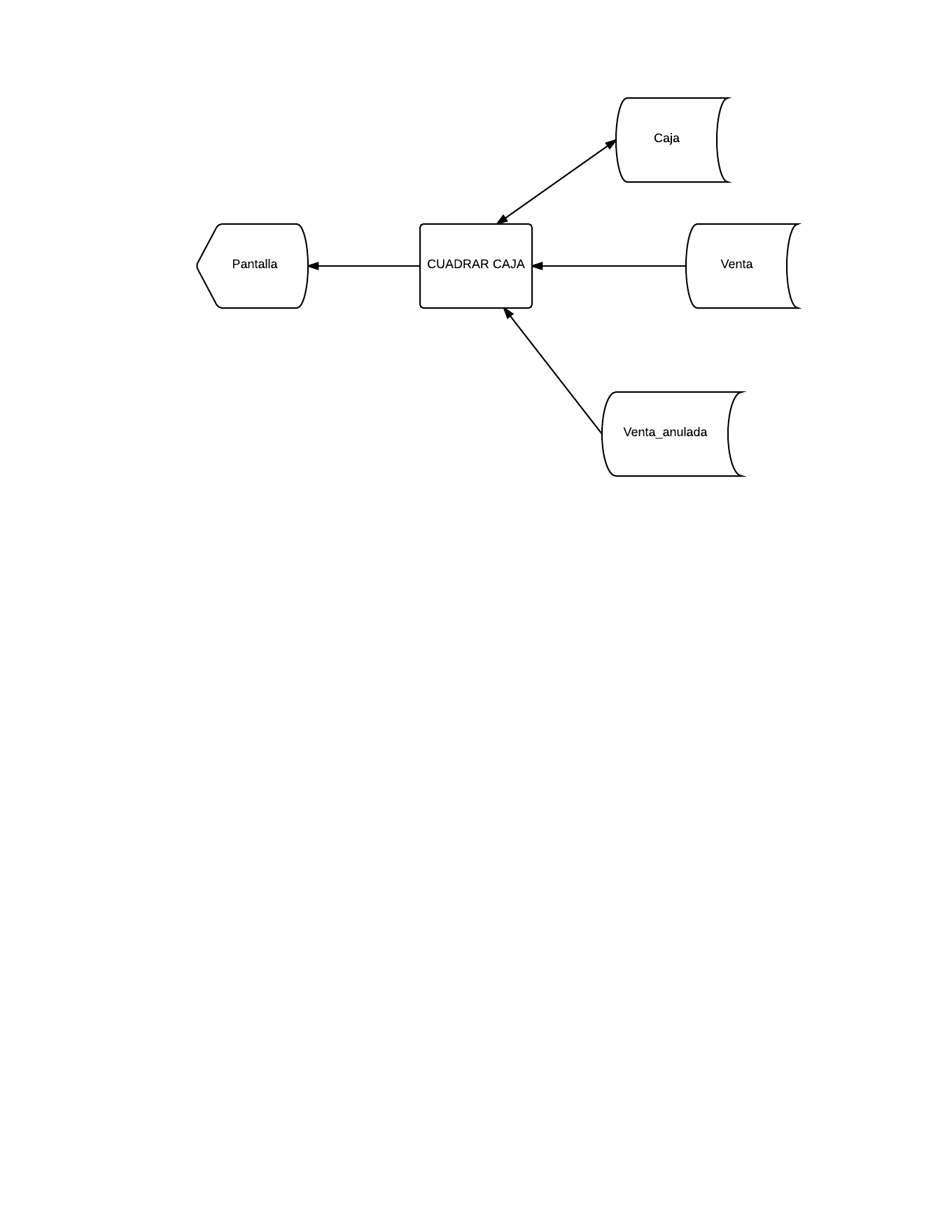


Fuente: Elaboración propia, basado en diseño de ventana de venta de la demostración

Figura 3-3 Diseño de pantalla venta

### 3.4.4. Cuadrar caja

* Nombre de programa: Cuadrar caja
* Objetivo: Registrar los movimientos monetarios computados (por el sistema) y reales (constatado por el trabajador) del cibercafé para una determinada sesión de un cierto cajero.
* Diagrama de bloque:



Fuente: Elaboración propia, basado en programa cuadrar caja

Diagrama 3-6. Diagrama de bloque cuadrar caja

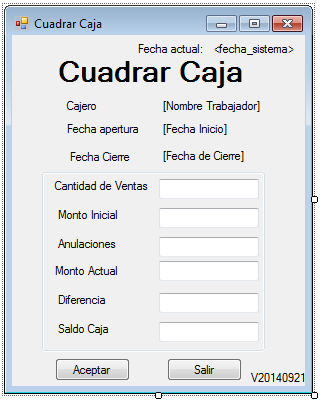
* Reglas de proceso

1. Al cargar el programa (haciendo clic en el botón “Cuadrar caja” del menú principal), el sistema verificará si el cajero ha abierto la caja en esta sesión.

Si la ha abierto, el sistema:

1. Recuperará el monto de apertura de la caja, buscando en la tabla CAJA donde el atributo *fecha\_hora\_inicio* coincida con la fecha de inicio de la sesión actual (almacenada localmente en una variable).
2. Obtiene el total recaudado por las ventas en la sesión, consultado la tabla VENTA.
3. Solicita a la base de datos la cantidad de ventas “concretadas” y la cantidad de ventas anuladas, utilizando las tablas VENTA Y VENTA\_ANULADA, respectivamente.
4. Calcula el monto actual computado (de acuerdo al sistema), restando el monto de apertura con el total de las ventas de la sesión.
5. Muestra los datos procesado y/o recuperado al usuario en la interfaz provista por el sistema.
6. A medida que el usuario llene la casilla “Saldo Caja”, con la cantidad de dinero que tenga la caja, el valor de “Diferencia” mostrará la desigualdad (si hay) entre el monto computado por el sistema y lo que realmente tiene la caja. “Diferencia” puede mostrar”:
   1. Cero (0) si la casilla está vacía o si los montos actuales coinciden.
   2. “?” si hay números negativos o caracteres inválidos.
   3. Algún número entero positivo si hay un valor correcto.
7. El sistema envía un mensaje al usuario para confirmar el monto de cierre de la caja
   1. Si lo acepta, se actualizan atributos de la tabla CAJA (donde CAJA.fecha\_hora\_inicio = fecha y hora de inicio de sesión del cajero). Se coloca la fecha y hora de cierre de la caja (esta corresponde al momento en el que el usuario confirma la operación) en *fecha\_hora\_termino* y se cierra el programa.
   2. Si el usuario no lo acepta entonces se cierra el cuadro de dialogo y se devuelve el control al programa.
8. En caso contrario, se mandará un mensaje de error al usuario y terminará la ejecución de esta rutina.

* Diseño de Pantalla

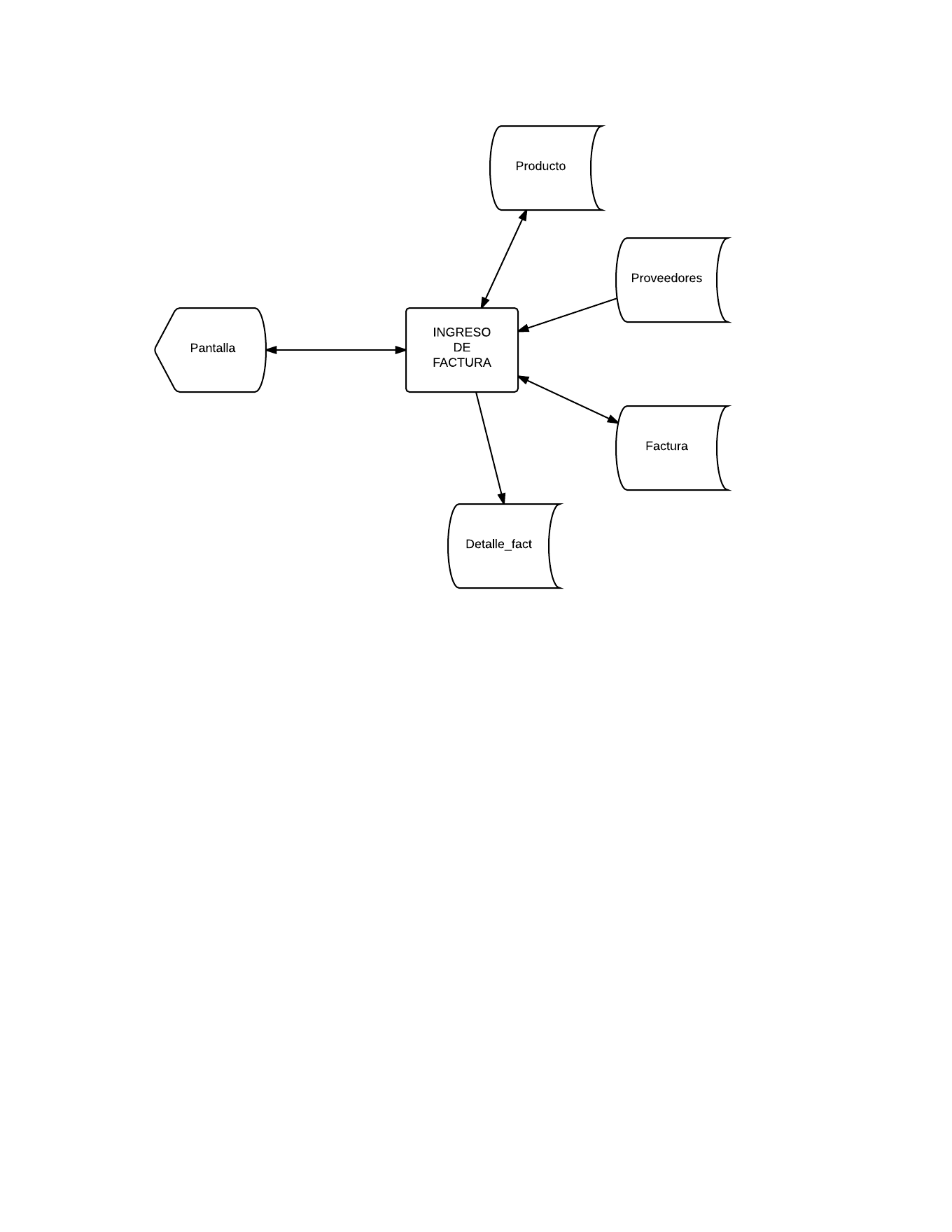


Fuente: Elaboración propia, basado en diseño de la ventana cuadrar caja de la demostración

Figura 3-4. Diseño de pantalla cuadrar caja

### 3.4.5. Ingreso de factura

* Nombre del programa: Ingreso de facturas
* Objetivo: Registrar el ingreso de productos adquiridos por el cibercafé.
* Diagrama de bloque:



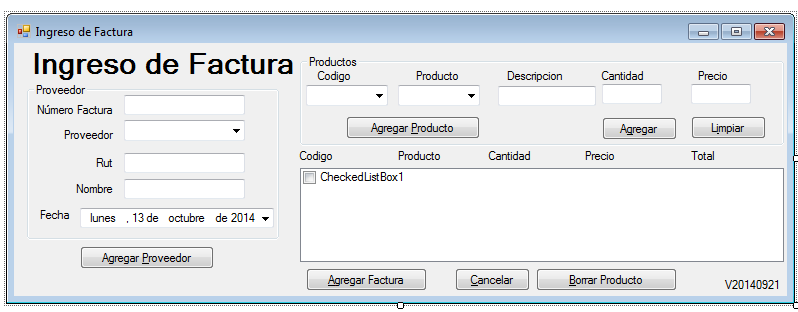
Fuente: Elaboración propia, basado en funcionamiento del proceso ingreso de factura

Diagrama 3-7. Diseño de bloque de ingreso de factura

* Reglas de proceso:

1. Al cargarse el programa, se consulta a:
   * 1. La tabla FACTURA, para obtener el número que tendrá la siguiente factura, correlativamente.
     2. La tabla de PROVEEDORES, para recuperar el nombre de cada uno de los proveedores, sus nombres de empresa y sus ruts asociados. Estos datos son poblados en una lista desplegable.
     3. La tabla PRODUCTO, de la cual se obtiene el código, nombre y precio de cada producto registrado. Estos datos son mostrados al usuario y son accesibles mediante una lista desplegable.
2. Si el usuario selecciona un proveedor de la lista desplegable, se le mostrará los datos asociados a dicho proveedor. Si no lo encontrare en la lista, puede ser agregado usando el botón “Agregar proveedores”. Esto lo llevará al programa “Mantenedor de proveedores”. Una vez que se cierre dicha ventana, el sistema actualizará la lista e incluirá el nuevo proveedor.
3. Si se selecciona un producto desde la combobox de productos, el usuario podrá ver más datos del artículo seleccionado. Al igual que con los proveedores, mediante el botón “Ingréselo”, puede incluirse nuevos proveedores a la factura actual.
4. Con “Agregar a la factura”, se adicionan productos a la factura. A medida que agregue, productos, el sistema calculará el total de la factura. Este es mostrado en “Total”, al lado derecho de la lista de productos. No se pueden agregar el mismo producto dos veces. Si la cantidad ingresada no tiene un número entero positivo, se considera como inválido, es rechazado y notificado al usuario mediante un mensaje de error.
5. “Borrar productos de la lista” permite eliminar uno o más productos de la lista de la factura, siempre que éstos estén seleccionados desde la lista (Si no hay ninguno seleccionado, se manda un mensaje de error).
6. Si se presiona el botón “Cancelar”, se cierra el programa registrador de facturas y se saca de memoria la lista de la factura que iba a ser ingresada (si hubiere alguna).
7. Al “Registrar la factura”, el algoritmo se cerciora de que:
   * 1. La factura tenga un producto asociado (si no lo hubiere, se mandaría un mensaje de error).
     2. Se incluya por lo menos un producto en la factura. (Si no tuviera ningún producto, se mandaría un mensaje de error).
8. Si todo lo anterior se cumple:
   * 1. Se crea un registro en la tabla FACTURA, que registra: el número de factura, el rut del proveedor involucrado, la fecha y hora de compra de los productos y el total.
     2. Se insertan, a la tabla DETALLE\_FACT, tantos registros como productos tenga la factura. Cada registro guarda el número de factura asociado, el rut del proveedor del que se adquirió dicho producto, cuántos productos fueron comprados, el precio unitario de compra y el subtotal (el total para cada artículo en particular).

* Diseño de Pantalla

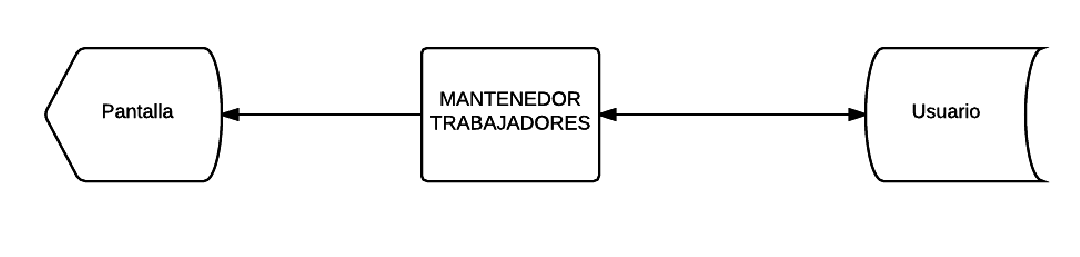


Fuente: Elaboración propia, basado en diseño de la ventana ingreso de factura de la demostración

Figura 3-5. Diseño de pantalla de ingreso de factura

### 3.4.6. Mantenedor de trabajadores

* Nombre del programa: Mantenedor de trabajadores.
* Objetivo: Buscar, agregar, actualizar y eliminar (datos de) trabajadores del cibercafé.
* Diagrama de bloque



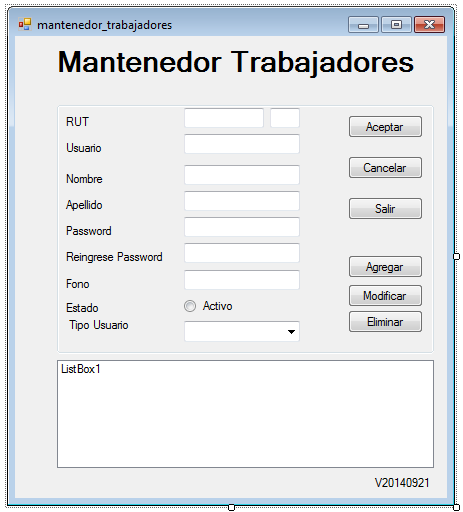
Fuente: Elaboración propia, basada en proceso mantenedor de trabajadores

Diagrama 3-8. Diagama de bloque mantenedor de trabajadores

* Reglas de proceso

1. Al cargarse el programa, se recuperan todas las tuplas de la tabla USUARIO, para ser mostradas al usuario en una lista.
2. Si el usuario presiona el botón “Agregar”, se esconden los botones que permiten modificar y eliminar usuarios, para después mostrar un formulario en el que se deben llenar los datos de un nuevo usuario a ser incluido en el sistema. De éstos, se valida que:
   * 1. Ninguno de los campos esté vacío.
     2. El rut ingresado sea válido.
     3. Las dos casillas de contraseña sean idénticas (*case-sensitive*).
     4. Que el fono sólo contenga: números, guiones “ – “ o paréntesis redondos “( )”
     5. Si los datos son válidos, se verifica en la tabla USUARIO que el rut ingresado no exista en el repositorio.
     6. Si no existe, se agrega un registro a la tabla mencionada, poblando la tupla con los suministrados. El tipo de usuario siempre es “Trabajador” (cod. Interno = 2)
     7. Además, el sistema envía un mensaje indicando que la transacción fue satisfactoria.
3. Mientras está agregándose un nuevo usuario, el usuario puede presionar “Cancelar” para ocultar el formulario de ingreso de nuevo usuario. Esto hará visibles los botones de eliminación y modificación de usuarios.
4. Si el usuario presiona el botón “Modificar”, se esconde los botones que agregar y eliminan trabajadores. Si:
   * 1. No se ha seleccionado ningún trabajador de la lista, se le mandara un mensaje de error indicándole que debe hacer esto para poder cambiar datos de algún funcionario.
     2. El usuario ha marcado alguno de los elementos de la lista y marca en “Modificar”, el sistema muestra los datos del trabajador seleccionado en un formulario con casillas modificables, salvo el rut, que no es alterable (es clave primaria de la tabla USUARIO).
     3. Si el usuario hace clic en “Aceptar”, el sistema buscará la tupla con los datos del usuario en cuestión (vía la clave primaria, *rut*), actualizará el registro y mostrará un cuadro de diálogo indicando que la transacción tuvo éxito. Después de que el usuario desaparezca el mensaje, volverá a la interfaz “inicial” del mantenedor de trabajadores.
     4. Si el usuario quiere “Cancelar” la actualización, se ocultarán las casillas con datos del trabajador en cuestión y se volverán a ver los botones para agregar y eliminar usuarios.
5. Si el usuario pulsa sobre el botón “Eliminar”,
   * 1. Recibirá un mensaje de error, por parte del sistema, en caso de que no haya seleccionado ningún trabajador antes de apretar dicho control.
     2. Se le pedirá que confirme si realmente desea eliminar el trabajador que seleccionó.
        1. Si responde “Sí”, entonces se busca al trabajador en la tabla USUARIO, por el rut, y se eliminar dicha tupla del repositorio.
        2. En caso contrario, que seleccione “no”, el cuadro de diálogo desaparecerá y podrá seguir trabajando con la interfaz normalmente.

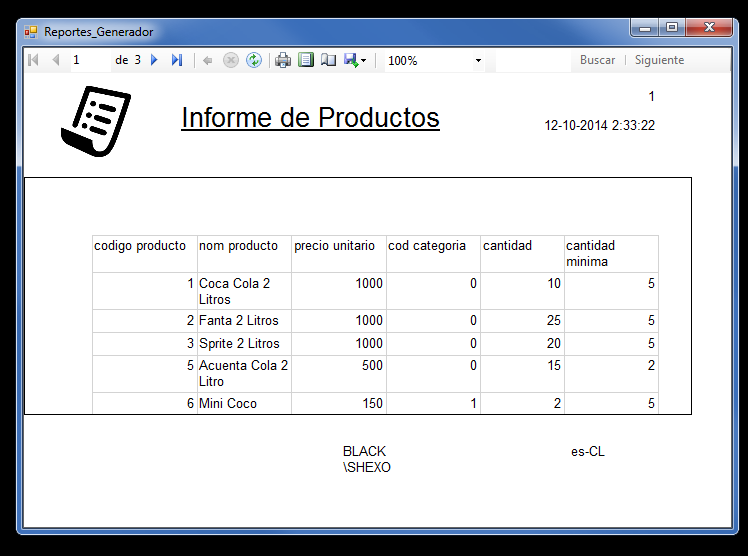
* Diseño de pantalla



Fuente: Elaboración propia, basado en diseño de la ventana mantenedor de la demostración

Figura 3-6. Diseño de pantalla mantenedor de trabajadores

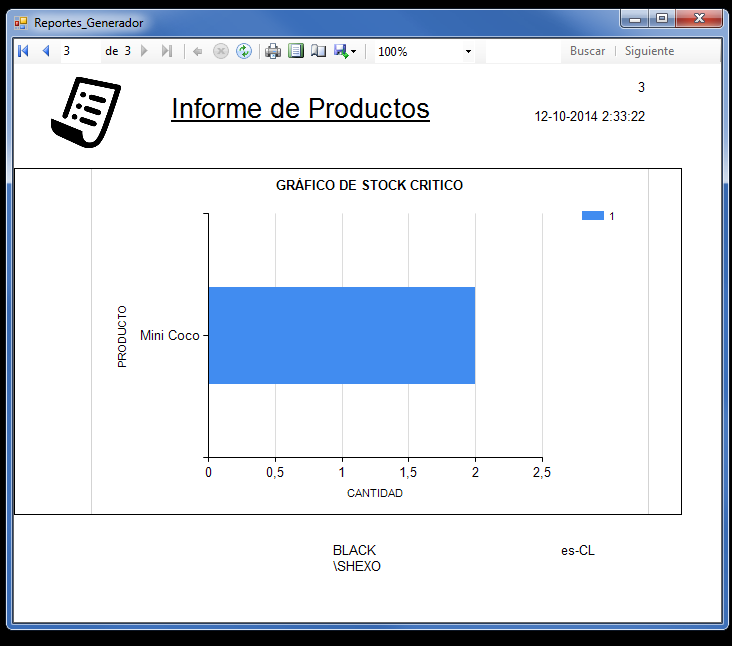
### 3.4.7. Informe de productos



Fuente: Elaboración propia, basado en tipo de informe de productos predeterminado

Figura 3-7. Diseño de pantalla de informe de productos

### 3.4.8. Informe de stock crítico

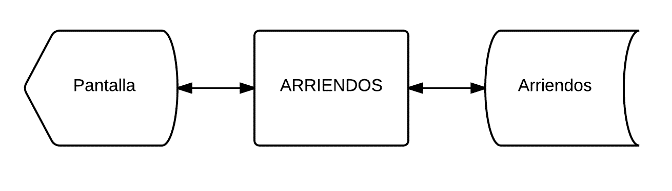


Fuente: Elaboración propia, basado en tipo de informe de productos predeterminado

Figura 3-8. Diseño de pantalla de informe de stock crítico

### 3.4.9. Arriendos

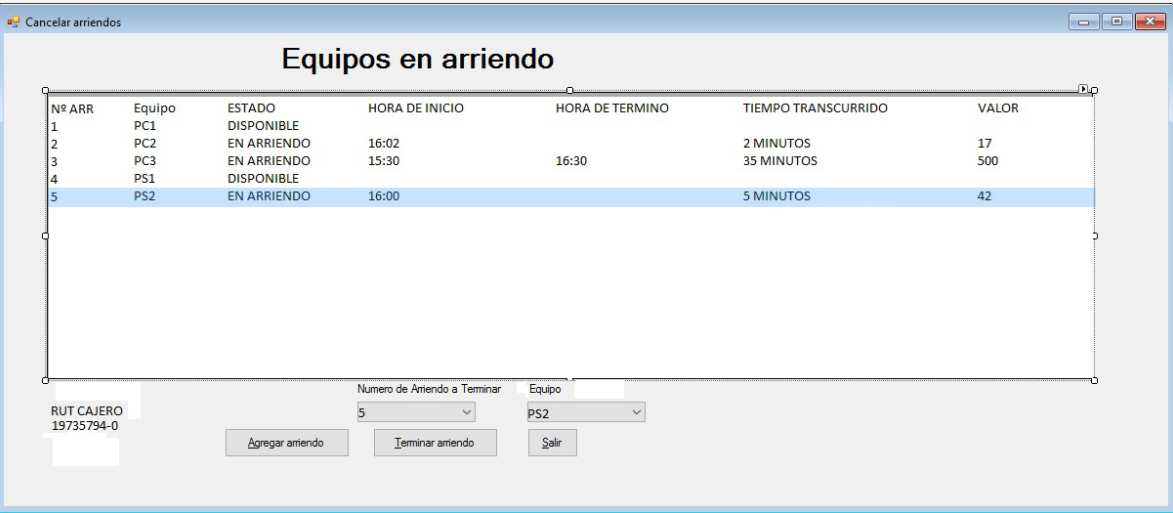
* Nombre de programa: Arriendos.
* Objetivo: Gestionar los arrendos de equipos del sistema del cibercafé.
* Diagrama de bloque:



Fuente: Elaboración propia, basado en proceso de arriendos

Diagrama 3-9. Diagrama de bloque arriendos

* Reglas de proceso
  + - 1. Se carga la ventana que muestra los equipos que estan registrados en el sistema, estén, o no, disponibles.
      2. Se busca en la tabla arriendo a los equipos que estén siendo arrendados. Para ellos, se muestra, en dos comboboxes, sus identificadores de equipo e identificador de arriendo, para que puedan ser seleccionados y dar fin a su arriendo.
      3. En la interfaz de usuario, se muestra, tambien, el rut del cajero que se ha autentificado en el sistema.
      4. Si se hace clic en "Terminar arriendo", el sistema mandará una consulta de insercion al DBMS para que se cambie el estado de disponibilidad del equipo seleccionado a disponible. Se enviará un mensaje mostrando el valor del arriendo a cancelar y el número de boleta del arriendo.
      5. Si escoge "Agregar arriendo", se cerrará la interfaz actual y se mostrará otra ventana que permite arrendar un equipo.
* Diseño de Pantalla

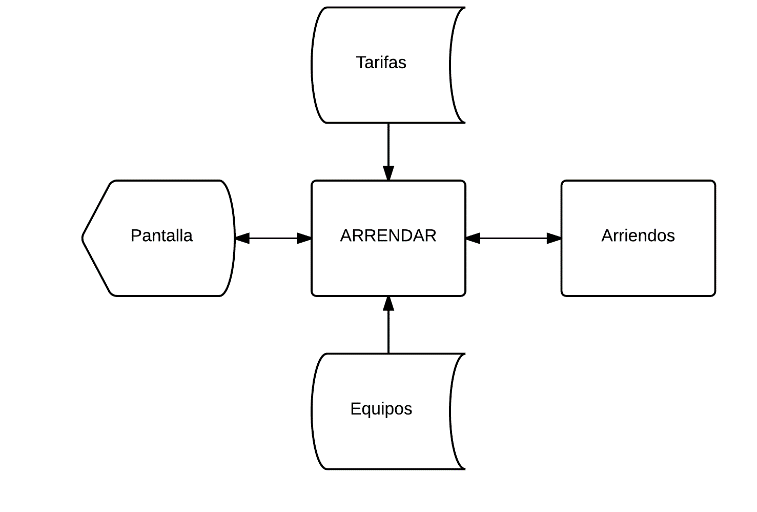


Fuente: Elaboración propia, basado en ventana de arriendos

Figura 3-9. Diseño de pantalla arriendos

### 3.4.10. Arrendar

* Nombre de programa: Arriendar
* Objetivo: Agrear un arriendo de un equipo al sistema del cibercafé.
* Diagrama de bloque:



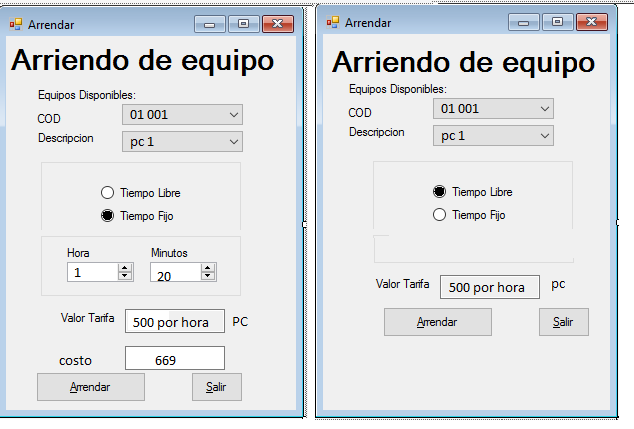
Fuente: Elaboración propia, basado en proceso de arrendar

Diagrama 3-10. Diagrama de bloque arrendar

* Reglas de proceso

1. Al cargar esta rutina, el sistema buscara en la base de datos el código identificador y la descripcion de los equipos que se encuentren "disponibles" para arrendar.
2. A medida que el usuario seleccione equipos, se ajustará la tarifa de arriendo, dependiendo, exclusivamente, del campo tipo\_equipo. Todos los datos mencionados son consultaods desde la tabla Equipos.
3. En cuanto el usuario ajuste la cantidad de tiempo por la que se realizará el arriendo, el sistema calculará la tarifa a cobrar.
4. Si se hace clic en "Salir", la rutina se cerrará y se volvera a la interfaz anterior: sea esta el menu principal o la ventana de control de equipo arrendados.
5. Si escoge "Arrendar", el usuario del sistema determina si se arrendará en modalidad de tiempo libre o tiempo fijo (discrminado por un radiobutton). En ambos casos, se inserta una tupla en la tabla arriendos del sistema, registrando datos del equipo y temporizadores. Se cambia el estado del equipo a 2 (arrendado). Ahora,
   1. Si se arrienda de tiempo libre, el campo de fecha y hora de termino queda nulo (vacio).
   2. Si no, se inserta la fecha y hora actual más el tiempo de arriendo solicitado (recogido desde la interfaz de usuario) y se envía mensaje de cobro del arriendo con su respectivo número de boleta002E.
6. Se envia un mensaje de confirmacion del arriendo y se devuelve el control al usuario.

* Diseño de Pantalla



Fuente: Elaboración propia, basado en ventana de arriendo de la demostración

Figura 3-10. Diseño de pantalla arrendar

# CONCLUSIONES Y RECOMENDACIONES

Luego de haber analizado, diseñado, confeccionado y probado el sistema en cuestión, hemos obtenido valiosa experiencia de cómo funciona realmente un cibercafé, más allá de lo que pueda percibir un simple usuario final. Por ejemplo, qué permite que un simple botón en una ventana de aplicación sea capaz, dentro de muchas otras posibilidades, de hacer que un equipo se bloquee a la distancia, sin siquiera tocarlo.

Este sistema nos ha ayudado, también, a desarrollar ciertas convenciones de ordenamiento lógico del código fuente (por ejemplo, de acuerdo al rol que cumplen) que desarrollamos. Así mismo, hemos procurado, dentro de lo posible, de lo conveniente, de lo prudente y de lo útil, reutilizar código y segmentarlo en módulos más sencillos de comprender y depurar, por ejemplo, al validar rangos numéricos o verificar ruts.

Ahora bien, el sistema posee algunos aspectos que podrían ser mejorados a futuro, a fin de entregar más o mejor funcionalidad, ofrecer un mejor rendimiento y hacer más fácil al usuario moverse por el sistema. Por ejemplo, podríamos haber escrito el sistema en C++ (que es procesado nativamente por Windows), en vez de Visual Basic (poseemos más conocimiento de éste último lenguaje. He ahí la razón de nuestra preferencia).

Uno de los puntos a mejorar es el manejo de las cadenas de conexión (actualmente almacenadas en una clase como variables de tipo cadena), que deberían ser movidas a un archivo de configuración del proyecto. Esto le daría independencia del código en sí.

Otro aspecto perfeccionable son las consultas realizadas al sistema gestor de bases de datos. Actualmente, hay muchas aperturas y cierres de conexión que podrían ser evitables. La conexión y desconexión con el DBMS implica un gran tiempo (perdido), ya que no sólo depende del motor, sino que también de la red e incluso del equipo que envía la consulta. Por otro lado, abrir y cerrar conexiones continuamente puede ponerle una carga innecesaria al DBMS.

Finalmente, el sistema desarrollado no será utilizado por el cibercafé, ya que la familia Sepúlveda dio por cerrado el cibercafé. Aun así es posible utilizar el sistema en cualquier otro cibercafé.

# BIBLIOGRAFÍA

* MSDN, Microsoft Developer Network [en línea] [Consulta: septiembre del 2014]. Disponible en :<social.msdn.microsoft.com>
* STACKOVERFLOW [en línea] [Consulta: agosto del 2014]. Disponible en <stackoverflow.com
* YOUTUBE [en línea] [Consulta: junio del 2014]. Disponible en: <www.[youtube.com](http://youtube.com/)>
* INGENERÍA DE SISTEMAS BLOG [en línea] [Consulta: agosto del 2014]. Disponible en :<eve-ingsistemas-u.blogspot.com.ar>
* MYSQL DEVELOPER ZONE [en línea] [Consulta: mayo del 2014]. Disponible en: <dev.mysql.com>
* MYSQL FORO [en línea] [Consulta: mayo del 2014]. Disponible en: <forums.mysql.com>
* ELHACKER.NET FORO [en línea] [Consulta: agosto del 2014]. Disponible en: <foro.elhacker.net>
* PROGRAMACIÓN CON ESTILO - BLOG [en línea] [Consulta: agosto del 2014]. Disponible en: <programacion-con-estilo.blogspot.com>
* TARINGA! - FORO [en línea] [Consulta: mayo del 2014]. Disponible en: <www.taringa.net>
* SOLO COGIDO BLOG [en línea] [Consulta: mayo del 2014]. Disponible en: <www.solocodigo.com>
* .NET HEAVEN FORO [en línea] [Consulta: octubre del 2014]. Disponible en: <www.dotnetheaven.com>
* GEEKSPROGRAMMINGS BLOG [en línea] [Consulta: mayo del 2014]. Disponible en: <geeksprogrammings.blogspot.com>
* C# CORNER [en línea] [Consulta: mayo del 2014]. Disponible en: <www.c-sharpcorner.com>
* VB.NET TUTORIAL AND SOURCE CODE [en línea] [Consulta: mayo del 2014]. Disponible en: <www.vb.net-informations.com>

# ANEXO A: ARCHIVOS DE PROGRAMA

/Login.vb

Public Class login

Private Sub login\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

lbl\_version.Text = "V" + Format(Date.Now(), "yyyyMMdd")

End Sub

Private Sub Btn\_Aceptar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn\_Aceptar.Click

Dim dao As New LoginDAO

Dim r As New Rut(TextBox1.Text)

If (validarIngreso(r)) Then

Try

If Not dao.existeUsuario(r.obtenerParteEntera & r.obtenerDV, TextBox2.Text) Then

MsgBox("El rut o contraseña son incorrectos", vbOK)

Else

ventana\_menu.Show()

Me.Visible = False

End If

Catch ex As MySql.Data.MySqlClient.MySqlException

MessageBox.Show("Error " & ex.Number.ToString & ": " & ex.Message)

End Try

End If

End Sub

Private Function validarIngreso(ByVal r As Rut) As Boolean

If TextBox1.Text = Nothing Then

MsgBox("Debe ingresar su RUT")

Return False

ElseIf Not r.rutValido() Then

MsgBox("El rut ingresado no es valido.")

Return False

ElseIf TextBox2.Text = Nothing Then

MsgBox("Debe ingresar su contraseña")

Else : Return True

End If

Return False

End Function

Sub Btn\_salir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_salir.Click

Me.Close()

End Sub

Private Sub lbl\_version\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles lbl\_version.Click

End Sub

End Class

/cuadrar\_caja.vb

Public Class cuadrar\_caja

Private fechaHoraEntradaAForm As String

Private fechaHoraInicioCaja As String

Private montoActual As Integer

Private saldoCaja As Integer

Private Sub formLoad(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Me.Load

If \_cajeroAbrioCaja Then

fechaHoraEntradaAForm = (Format(DateTime.Now(), "yyyy-MM-dd HH:mm:ss"))

lbl\_muestra\_fecha.Text = Format(Now, "short Date")

Label11.Text = \_cajeroNombre & " " & \_cajeroApellidos

Dim dao As CajaDAO = New CajaDAO

'1.1 Obtener la fechaHora de inicio de la caja ( y no de la persona que inicio sesion)

'Format(DateTime.Now, "yyyy-MM-dd HH:mm:ss")

'Dim fechaHoraInicioCaja As String = dao.obtenerFechaApertura

fechaHoraInicioCaja = \_cajeroFechaAperturaCaja

Label12.Text = fechaHoraInicioCaja

'1.2 Obtener monto inical de la caja

Dim montoInicial As Integer = dao.obtenerMontoApertura()

'2.Calcular total de ventas para la sesion

Dim totalVentas As Integer = dao.obtenerTotalVentasSesionTrab(fechaHoraInicioCaja, fechaHoraEntradaAForm)

'2.1 Obtener nro de ventas anuladas y no anuladas

Dim cantVentas As Integer = dao.obtenerCantVentasSesion(fechaHoraInicioCaja, fechaHoraEntradaAForm)

Dim cantAnuladas As Integer = dao.obtenerCantVentasAnuladas(fechaHoraInicioCaja, fechaHoraEntradaAForm)

'3. Calcular Monto acutal como: monto inicial - cant ventas

montoActual = montoInicial - totalVentas

'4. Mostrarle los datos al Sr. Usuario

lbl\_anulaciones.Text = cantAnuladas

lbl\_cant\_ventas.Text = cantVentas

lbl\_monto\_inicial.Text = montoInicial

lbl\_monto\_actual.Text = montoActual

Else

MsgBox("La caja aun no ha sido abierta", vbOKOnly)

Me.Close()

End If

End Sub

Private Sub btn\_salir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_salir.Click

Me.Close()

End Sub

Private Sub txt\_saldo\_caja\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txt\_saldo\_caja.TextChanged

Dim c As CodigosNumericos = New CodigosNumericos(txt\_saldo\_caja.Text)

If txt\_saldo\_caja.Text = Nothing Then

lbl\_diferencia.Text = 0

ElseIf c.validarCodigo() Then

saldoCaja = txt\_saldo\_caja.Text

Dim diferencia As Integer = lbl\_monto\_actual.Text - saldoCaja

lbl\_diferencia.Text = diferencia

Else : lbl\_diferencia.Text = "?"

End If

End Sub

Private Sub btn\_cuadrar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_cuadrar.Click

Dim c As New CodigosNumericos(txt\_saldo\_caja.Text)

If Not c.validarCodigo Then

MsgBox("Elk valor ingresado como saldo de caja no es valido" & vbCrLf & "¿Esta seguro/a de que solo contiene numeros?", vbOKOnly, vbInformation)

ElseIf MsgBox("¿Esta seguro de querer cerrar la caja con $" & txt\_saldo\_caja.Text & "?", vbYesNo) = vbYes Then

Dim dao As New CajaDAO

If dao.cerrarCaja(montoActual, saldoCaja) = 1 Then

MsgBox("La caja ha sido cerrada exitosamente")

Else

MsgBox("Ocurrio un error al cerrar la caja. Porfavor, contactese con el administrador")

End If

End If

End Sub

End Class

/ingreso\_factura.vb

Public Class ingreso\_factura

Private prodSeleccionado As New ProductoVO

Private total As Double = 0

Private source As New DataTable

Private codigosIngresados As ArrayList = New ArrayList

Private Sub formLoad(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim dao As FacturaDAO = New FacturaDAO

txt\_nro\_fact.Text = dao.sugerirNroFact()

Me.llenarComboboxProveedores()

llenarComboboxProductos()

inicializarTablaProductos()

End Sub

Private Sub Button5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click

mantenedor\_proveedores.Show()

End Sub

Private Sub btn\_registrar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_registrar.Click

If validarIngresoFactura() Then

If MsgBox("¿Está seguro de querer ingresar la factura con los productos de la lista?", vbYesNo, vbQuestion) = vbYes Then

Dim fact As FacturaVO = New FacturaVO

Dim rut\_proveedor As String = lbl\_rut\_proveedor.Text

rut\_proveedor = rut\_proveedor.Replace("-", "")

DateTimePicker1.Format = DateTimePickerFormat.Custom

DateTimePicker1.CustomFormat = "yyyy-MM-dd HH:mm:ss"

Dim numero\_factura As Integer = Convert.ToInt32(txt\_nro\_fact.Text)

With fact

.setRut\_proveedor(rut\_proveedor)

.setNumero\_factura(numero\_factura)

.setFecha\_compra(DateTimePicker1.Text)

.setTotal(Convert.ToInt32(total))

End With

Dim lista As ArrayList = New ArrayList

For Each row As DataGridViewRow In DataGridView1.Rows

Dim detalle As Detalle\_FactVO = New Detalle\_FactVO

With detalle

.setCod\_producto(Convert.ToInt32(row.Cells("Codigo").Value))

.setNumero\_factura(numero\_factura)

.setRut\_proveedor(rut\_proveedor)

.setCant\_comprada(Convert.ToInt32(row.Cells("Cantidad").Value))

.setPrecio\_unit\_compra(Convert.ToInt32(row.Cells("Precio").Value))

.setSubtotal(Convert.ToInt32(row.Cells("Subtotal").Value))

End With

lista.Add(detalle)

Next

MsgBox(rut\_proveedor)

Dim fact\_dao As FacturaDAO = New FacturaDAO

Dim det\_dao As Detalle\_FactDAO = New Detalle\_FactDAO

fact\_dao.agregar(fact)

det\_dao.agregar(lista)

Else : MsgBox("Hay datos mal ingresados.")

End If

End If

End Sub

Private Sub Button7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button7.Click

mantenedor\_productos.Show()

End Sub

Private Sub cmb\_nombre\_proveedor\_SelectedValueChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles cmb\_nombre\_proveedor.SelectedValueChanged

Dim dao As New FacturaDAO

Dim vo As New ProveedoresVO

vo = dao.buscarProvPorNombre(cmb\_nombre\_proveedor.SelectedItem)

lbl\_nombre\_empresa.Text = vo.getNombre\_empresa\_proveedor

Dim rut As String = vo.getRut\_proveedor

lbl\_rut\_proveedor.Text = String.Format("{0:###.###.###}", rut.Substring(0, rut.ToString.Length - 1)) & "-" & rut.Substring(rut.Length - 1)

End Sub

Private Sub llenarComboboxProveedores()

Me.cmb\_nombre\_proveedor.Items.Clear()

Dim dao As New FacturaDAO

Dim lista As ArrayList = dao.obtenerProveedores()

For Each vo As ProveedoresVO In lista

cmb\_nombre\_proveedor.Items.Add(vo.getNombre\_proveedor)

Next

End Sub

Private Sub btn\_salir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_salir.Click

Me.Close()

End Sub

Private Sub llenarComboboxProductos()

Me.cmb\_codigos.Items.Clear() 'para no tener datos repetidos al recargar

Dim dao As New FacturaDAO

Dim lista As ArrayList = dao.obtenerCodigos()

For Each vo As ProductoVO In lista

cmb\_codigos.Items.Add(vo.getCod\_producto & " - " & vo.getNom\_producto)

Next

End Sub

Private Sub cmb\_codigos\_SelectedValueChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles cmb\_codigos.SelectedValueChanged

Dim dao As New FacturaDAO

Dim vo As ProductoVO = dao.buscarProdPorCodigo(cmb\_codigos.SelectedItem)

lbl\_nom\_producto.Text = vo.getNom\_producto()

lbl\_precio\_unitario.Text = vo.getPrecio\_unitario()

lbl\_cod\_producto.Text = vo.getCod\_producto()

prodSeleccionado = vo

End Sub

Private Sub btn\_agregar\_prod\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_agregar\_prod.Click

If cmb\_codigos.SelectedIndex = -1 Then

MsgBox("Seleccione un producto a agregar")

cmb\_codigos.Focus()

Else

Try

Dim cantidad As Integer = Convert.ToInt32(txt\_cantidad.Text)

If cantidad <= 0 Then

MsgBox("La cantidad a ingresar no puede ser cero o menor.")

txt\_cantidad.Focus()

Else

Dim codigo As Integer = prodSeleccionado.getCod\_producto

If codFueIngresado(codigosIngresados, codigo) Then

MsgBox("El producto ya existe en la lista")

Exit Sub

End If

codigosIngresados.Add(codigo)

Dim producto As String = prodSeleccionado.getNom\_producto

Dim precio As Integer = prodSeleccionado.getPrecio\_unitario

Dim subtotal As Double = precio \* cantidad

total = total + subtotal

lbl\_total.Text = total

Dim row As Object() = New Object() {codigo, producto, cantidad, precio, subtotal}

DataGridView1.Rows.Add(row)

DataGridView1.Show()

DataGridView1.ClearSelection()

End If

Catch ex As FormatException

MsgBox("La cantidad solo puede llevar numeros")

End Try

End If

End Sub

Private Sub inicializarTablaProductos()

DataGridView1.Columns.Add("Codigo", "Codigo")

DataGridView1.Columns.Add("Producto", "Producto")

DataGridView1.Columns.Add("Cantidad", "Cantidad")

DataGridView1.Columns.Add("Precio", "Precio")

DataGridView1.Columns.Add("Subtotal", "Subtotal")

DataGridView1.Hide()

DataGridView1.Columns("Codigo").Frozen = True

DataGridView1.Columns("Producto").ReadOnly = True

DataGridView1.Columns("Cantidad").ReadOnly = True

DataGridView1.Columns("Precio").ReadOnly = True

DataGridView1.Columns("Subtotal").ReadOnly = True

'// Inicalizacion dle DataSet

source.Columns.Add("Codigo", Type.GetType("System.Int32"))

source.Columns.Add("Producto", Type.GetType("System.String"))

source.Columns.Add("Cantidad", Type.GetType("System.Int32"))

source.Columns.Add("Precio", Type.GetType("System.Int32"))

source.Columns.Add("Subtotal", Type.GetType("System.Double"))

DataGridView1.AutoGenerateColumns = False

End Sub

Private Sub actualizarDGV()

DataGridView1.DataSource = source

End Sub

Private Sub btn\_borrar\_prod\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_borrar\_prod.Click

If Me.DataGridView1.SelectedRows.Count = 0 Then

MsgBox("Para eliminar productos, marquelos desde la parte izquerda de la lista")

Else

Dim totalResta As Double = 0

For i = Me.DataGridView1.SelectedRows.Count - 1 To 0 Step -1

Dim subtotalFila As Integer = Convert.ToUInt32(Me.DataGridView1.SelectedCells.Item(Me.DataGridView1.Columns("Subtotal").Index).Value)

subtotalFila \*= -1

totalResta = totalResta - subtotalFila

Me.DataGridView1.Rows.Remove(Me.DataGridView1.SelectedRows(i))

Next

total = total - totalResta

lbl\_total.Text = total

DataGridView1.ClearSelection()

End If

End Sub

Private Function validarIngresoFactura() As Boolean

Dim c As CodigosNumericos = New CodigosNumericos(txt\_nro\_fact.Text)

If txt\_nro\_fact.Text = Nothing Then

MsgBox("La factura debe llevar un numero")

ElseIf (Not c.validarCodigo()) Then

MsgBox("El codigo debe contener caracteres (solo numericos).")

txt\_nro\_fact.Focus()

ElseIf cmb\_codigos.SelectedIndex = -1 Then

MsgBox("Debe seleccionar un proveedor.")

cmb\_codigos.Focus()

ElseIf DataGridView1.Rows.Count <= 0 Then

MsgBox("La factura debe tener, por lo menos, un producto")

cmb\_nombre\_proveedor.Focus()

Else : Return True

End If

Return False

End Function

Private Function codFueIngresado(ByVal lista As ArrayList, ByVal nuevoCodigo As Integer)

For i = 0 To lista.Count - 1 Step 1

If lista(i) = nuevoCodigo Then

Return True

End If

Next

Return False

End Function

Private Sub Form1\_Activated(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Activated

Me.llenarComboboxProveedores()

Me.llenarComboboxProductos()

End Sub

End Class

/mantenedor\_trabajadores

Imports System.Text.RegularExpressions

Public Class mantenedor\_trabajadores

Private agrega As Boolean = False

Private valido As Boolean = True

Private usuarioAnt As String

Private retornar As Boolean = False

Dim dao As New TrabajadorDAO

Private Sub formLoad(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Try

poblarDataGridView1()

GroupBox1.Hide()

lbl\_interplay.Show()

btn\_agregar.Focus()

btn\_cancelar.Hide()

btn\_aceptar.Hide()

Catch ex As Exception

MsgBox("Ocurrio un error al conectar con la base de datos.")

MsgBox(ex.StackTrace)

Me.Close()

End Try

End Sub

Private Sub btn\_salir\_click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_salir.Click

Me.Close()

End Sub

Private Sub DataGridView1\_CellSingleClick(sender As Object, e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellClick

If e.RowIndex >= 0 Then

Dim fila As DataGridViewRow

fila = Me.DataGridView1.Rows(e.RowIndex)

Dim rut As String = fila.Cells("Rut").Value.ToString

txt\_rut.Text = rut.Substring(0, rut.Length - 1)

txt\_dv.Text = rut.Substring(Len(rut) - 1, 1)

txt\_nombre.Text = fila.Cells("Nombre").Value.ToString

txt\_apellido.Text = fila.Cells("Apellidos").Value.ToString

txt\_pass.Text = fila.Cells("Password").Value.ToString

txt\_reingrese\_pass.Text = txt\_pass.Text

txt\_fono.Text = fila.Cells("Fono").Value.ToString

If fila.Cells("Estado").Value = 0 Then 'está activo

chk\_estado.Checked = True

Else

chk\_estado.Checked = False

End If

End If

End Sub

Private Sub btn\_modificar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_modificar.Click

If txt\_rut.Text = Nothing Or txt\_dv.Text = Nothing Then

MsgBox("Debe seleccionar un trabajador.")

Else

lbl\_interplay.Hide()

GroupBox1.Show()

agrega = False

txt\_rut.ReadOnly = True

txt\_dv.ReadOnly = True

btn\_eliminar.Hide()

btn\_agregar.Hide()

btn\_cancelar.Show()

btn\_aceptar.Show()

End If

End Sub

Private Sub btn\_agregar\_Click(sender As System.Object, e As System.EventArgs) Handles btn\_agregar.Click

lbl\_interplay.Hide()

GroupBox1.Show()

agrega = True

btn\_eliminar.Hide()

btn\_modificar.Hide()

lbl\_estado.Hide()

chk\_estado.Hide()

btn\_cancelar.Show()

btn\_aceptar.Show()

End Sub

Private Sub btn\_eliminar\_Click(sender As System.Object, e As System.EventArgs) Handles btn\_eliminar.Click

If txt\_rut.Text = Nothing Then

MsgBox("Debe seleccionar un trabajador")

Else

If MsgBox("¿Esta seguro que desea eliminar al trabajador <" & txt\_nombre.Text & " " & txt\_apellido.Text & ">?", vbYesNo) = vbYes Then

Dim rut As String = txt\_rut.Text.ToString & txt\_dv.Text.ToString 'rut sin formato

dao.eliminar(rut)

End If

End If

poblarDataGridView1()

End Sub

Private Sub btn\_cancelar\_Click(sender As System.Object, e As System.EventArgs) Handles btn\_cancelar.Click

If Not agrega Then

txt\_rut.ReadOnly = False

txt\_dv.ReadOnly = False

End If

GroupBox1.Hide()

lbl\_interplay.Show()

btn\_eliminar.Show()

btn\_agregar.Show()

btn\_modificar.Show()

lbl\_estado.Show()

chk\_estado.Show()

btn\_cancelar.Hide()

btn\_aceptar.Hide()

Me.limpiarCampos()

End Sub

Private Sub btn\_aceptar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_aceptar.Click

retornar = False

If Me.validarCampos Then

If agrega Then

If Not dao.buscarPorRut(txt\_rut.Text & txt\_dv.Text) Then

Dim estado As Integer = 0

Dim vo As New TrabajadorVO

vo.setRut(txt\_rut.Text.ToString & txt\_dv.Text.ToString)

vo.setNombre(txt\_nombre.Text.ToString)

vo.setApellido(txt\_apellido.Text.ToString)

vo.setPassword(txt\_pass.Text)

vo.setFono(Val(txt\_fono.Text))

vo.setEstado(estado)

dao.agregar(vo)

retornar = True

Else : MsgBox("El rut que ingreso ya existe en el sistema.")

End If

ElseIf Not agrega Then 'Modifica;

Dim estado As Integer

If chk\_estado.Checked Then

estado = 0

Else : estado = 1

End If

Dim vo As New TrabajadorVO

vo.setRut(txt\_rut.Text.ToString & txt\_dv.Text.ToString)

vo.setNombre(txt\_nombre.Text.ToString)

vo.setApellido(txt\_apellido.Text.ToString)

vo.setPassword(txt\_pass.Text)

vo.setFono(Val(txt\_fono.Text))

vo.setEstado(estado)

dao.actualizar(vo)

retornar = True

End If

If retornar Then 'Solo si el usuario logro insertar/act. existosamente cambio la visual

poblarDataGridView1()

GroupBox1.Hide()

lbl\_interplay.Show()

btn\_eliminar.Show()

btn\_agregar.Show()

btn\_modificar.Show()

lbl\_estado.Show()

chk\_estado.Show()

btn\_eliminar.Show()

btn\_agregar.Show()

Me.limpiarCampos()

End If

End If

End Sub

Private Function validarCampos() As Boolean

Dim r As New Rut(txt\_rut.Text & "-" & txt\_dv.Text)

Dim fono As New Fono(txt\_fono.Text)

If txt\_rut.Text = Nothing Or txt\_dv.Text = Nothing Then

MsgBox("Debe ingresar su rut")

ElseIf Not r.rutValido() Then

MsgBox("El rut ingresado no es valido.")

ElseIf txt\_nombre.Text = Nothing Then

MsgBox("Debe ingresar su nombre.")

ElseIf txt\_apellido.Text = Nothing Then

MsgBox("Debe ingresar su apellido.")

ElseIf Not txt\_pass.Text.Equals(txt\_reingrese\_pass.Text) Then

MsgBox("Las contraseñas ingresadas no coinciden.")

ElseIf Not fono.validarFono() Then

MsgBox("El telefono ingresado contiene caracteres invalidos.")

Else

Return True

End If

Return False

End Function

Private Sub poblarDataGridView1()

Dim dao As New TrabajadorDAO

DataGridView1.DataSource = dao.consultarTodos.Tables("usuario")

DataGridView1.ClearSelection()

End Sub

Private Sub limpiarCampos()

Me.txt\_apellido.Text = Nothing

Me.txt\_dv.Text = Nothing

Me.txt\_fono.Text = Nothing

Me.txt\_nombre.Text = Nothing

Me.txt\_pass.Text = Nothing

Me.txt\_reingrese\_pass.Text = Nothing

Me.txt\_rut.Text = Nothing

End Sub

Private Sub DataGridView1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick

End Sub

End Class

/venta.vb

Public Class venta

Private Sub venta\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Me.llenarComboboxCodigos()

lbl\_venta.Text = DateAndTime.Now

cmb\_productos.SelectedIndex = 0

llenarComboboxVenta()

cmb\_venta.SelectedIndex = 0

lbl\_venta.Text = cmb\_venta.Text '

lbl\_rut.Text = \_cajeroRut

End Sub

Private Sub btn\_salir\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_salir.Click

Me.Close()

End Sub

Public Sub llenarComboboxVenta()

Me.cmb\_venta.Items.Clear() 'para no tener datos repetidos al recargar

Dim dao As New VentaDAO

Dim lista As ArrayList = dao.recuperarCodigos()

For Each vo As VentaVO In lista

cmb\_venta.Items.Add(vo.getNumero\_venta)

Next

End Sub

Public Sub llenarComboboxCodigos()

Me.cmb\_codigo.Items.Clear() 'para no tener datos repetidos al recargar

Dim dao As New ProductoDAO

Dim lista As ArrayList = dao.recuperarCodigos()

For Each vo As ProductoVO In lista

cmb\_codigo.Items.Add(vo.getCod\_producto)

cmb\_productos.Items.Add(vo.getNom\_producto)

cmb\_precio.Items.Add(vo.getPrecio\_unitario)

cmb\_cantidad.Items.Add(vo.getCantidad)

num\_cantidad.Maximum = vo.getCantidad

Next

End Sub

Private Sub cmb\_productos\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmb\_productos.SelectedIndexChanged

cmb\_codigo.SelectedIndex = cmb\_productos.SelectedIndex

cmb\_precio.SelectedIndex = cmb\_productos.SelectedIndex

cmb\_cantidad.SelectedIndex = cmb\_productos.SelectedIndex

txt\_precio.Text = cmb\_precio.Text

num\_cantidad.Maximum = cmb\_cantidad.Text

End Sub

Private Sub cmb\_codigo\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmb\_codigo.SelectedIndexChanged

cmb\_productos.SelectedIndex = cmb\_codigo.SelectedIndex

cmb\_precio.SelectedIndex = cmb\_codigo.SelectedIndex

cmb\_cantidad.SelectedIndex = cmb\_productos.SelectedIndex

txt\_precio.Text = cmb\_precio.Text

num\_cantidad.Maximum = cmb\_cantidad.Text

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

DataGridView1.Rows.Add(cmb\_codigo.Text, cmb\_productos.Text, num\_cantidad.Text, txt\_precio.Text \* num\_cantidad.Text)

txt\_subtotal.Text = CDbl(txt\_precio.Text) \* CDbl(num\_cantidad.Text) + CDbl(txt\_subtotal.Text)

txt\_total = txt\_subtotal

If DataGridView1.RowCount > 0 Then

btn\_vender.Enabled = True

End If

End Sub

Private Sub btn\_borrarLista\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_borrarLista.Click

DataGridView1.Rows.Clear()

txt\_subtotal.Text = "0"

txt\_total = txt\_subtotal

btn\_vender.Enabled = False

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

If Not (DataGridView1.Rows(DataGridView1.CurrentRow.Index) Is DBNull.Value) Then \_

DataGridView1.Rows.RemoveAt(DataGridView1.CurrentRow.Index)

End Sub

Private Sub btn\_vender\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_vender.Click

vender()

actualizarInventario()

Me.Close()

End Sub

Private Sub vender()

Dim dao As New VentaDAO

Dim vo As New VentaVO

vo.setNumero\_venta(lbl\_venta.Text)

vo.setTotal(txt\_total.Text)

vo.setRut\_trabajador(lbl\_rut.Text)

vo.setTipo\_venta("1")

vo.setARRIENDO\_numero\_arriendo("0")

dao.agregar(vo)

Dim StrSQL As String = ""

If DataGridView1.RowCount > 0 Then

For Each Dt\_Row As DataGridViewRow In DataGridView1.Rows

Dim daoDetalle As New Detalle\_VentaDAO

Dim voDetalle As New Detalle\_VentaVO

voDetalle.setCod\_producto(Dt\_Row.Cells(0).Value)

voDetalle.setNumero\_venta(lbl\_venta.Text)

voDetalle.setCantidad(Dt\_Row.Cells(2).Value)

daoDetalle.agregar(voDetalle)

Next

End If

MsgBox("Venta Realizada")

End Sub

Private Sub actualizarInventario()

Dim StrSQL As String = ""

For Each Dt\_Row As DataGridViewRow In DataGridView1.Rows

Dim dao As New ProductoDAO

Dim vo As New ProductoVO

vo.setCod\_producto(Dt\_Row.Cells(0).Value)

vo.setCantidad(Dt\_Row.Cells(2).Value)

dao.actualizar\_por\_venta(vo)

Next

End Sub

Private Sub GroupBox1\_Enter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GroupBox1.Enter

End Sub

End Class

/ventana\_menu

Public Class ventana\_menu

Dim \_enabledCerrar As Boolean = False

<System.ComponentModel.DefaultValue(False), System.ComponentModel.Description("Define si se habilita el botón cerrar en el formulario")> \_

Public Property EnabledCerrar() As Boolean

Get

Return \_enabledCerrar

End Get

Set(ByVal Value As Boolean)

If \_enabledCerrar <> Value Then

\_enabledCerrar = Value

End If

End Set

End Property

Protected Overrides ReadOnly Property CreateParams() As CreateParams

Get

Dim cp As CreateParams = MyBase.CreateParams

If \_enabledCerrar = False Then

Const CS\_NOCLOSE As Integer = &H200

cp.ClassStyle = cp.ClassStyle Or CS\_NOCLOSE

End If

Return cp

End Get

End Property

Private Sub ventana\_menu\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

lbl\_version.Text = "V" + Format(Now, "yyyyMMdd")

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

apertura\_caja.Show()

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

venta.Show()

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

ajustar\_stock.Show()

End Sub

Private Sub Button4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

anular\_venta.Show()

End Sub

Private Sub Button8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button8.Click

asignar\_categoria\_producto.Show()

End Sub

Private Sub Button7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button7.Click

consultar\_stock.Show()

End Sub

Private Sub Button6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click

cuadrar\_caja.Show()

End Sub

Private Sub Button5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click

ingreso\_factura.Show()

End Sub

Private Sub Button12\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button12.Click

mantenedor\_categoria.Show()

End Sub

Private Sub Button11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button11.Click

mantenedor\_clientes.Show()

End Sub

Private Sub Button10\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button10.Click

mantenedor\_productos.Show()

End Sub

Private Sub Button9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button9.Click

mantenedor\_proveedores.Show()

End Sub

Private Sub Button13\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button13.Click

mantenedor\_trabajadores.Show()

End Sub

Private Sub Button14\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button14.Click

MessageBox.Show("Se conectará a " & txt\_cadena.Text)

mantenedor\_equipos.Show()

End Sub

Private Sub Button15\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button15.Click

arriendos.Show()

End Sub

Private Sub Button16\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button16.Click

Reportes\_Generador.Show()

End Sub

Private Sub VentaToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles VentaToolStripMenuItem.Click

venta.Show()

End Sub

Private Sub SalirToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SalirToolStripMenuItem.Click

Dim dao As New HistorialSesionesTrabDAO

If Not dao.cerrarSesion() Then

MsgBox("Ups... Ocurrio un error al guardar el fin de sesión. Porfavor comuniquese con el administrador")

End If

End

End Sub

Private Sub MantendorDeEquiposToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MantendorDeEquiposToolStripMenuItem.Click

mantenedor\_equipos.MdiParent = Me

mantenedor\_equipos.Show()

End Sub

Private Sub Button17\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button17.Click

consultar\_stock.Show()

End Sub

Private Sub ArriendoToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

arriendos.Show()

End Sub

Private Sub MantendorTrabajadoresToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MantendorTrabajadoresToolStripMenuItem.Click

mantenedor\_trabajadores.Show()

End Sub

Private Sub AperturaCajaToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AperturaCajaToolStripMenuItem1.Click

apertura\_caja.Show()

End Sub

Private Sub CuadrarCajaToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CuadrarCajaToolStripMenuItem1.Click

cuadrar\_caja.Show()

End Sub

Private Sub ConsultarStockToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ConsultarStockToolStripMenuItem.Click

consultar\_stock.Show()

End Sub

Private Sub MantenedorClientesToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MantenedorClientesToolStripMenuItem.Click

mantenedor\_clientes.Show()

End Sub

Private Sub MantenedorProveedoresToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MantenedorProveedoresToolStripMenuItem.Click

'MessageBox.Show("En Desarrollo")

mantenedor\_proveedores.Show()

End Sub

Private Sub MantendorCategoriaToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MantendorCategoriaToolStripMenuItem.Click

mantenedor\_categoria.Show()

End Sub

Private Sub AnularVentaToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AnularVentaToolStripMenuItem.Click

anular\_venta.Show()

End Sub

Private Sub ConsultarProductoToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ConsultarProductoToolStripMenuItem.Click

consultar\_stock.Show()

End Sub

Private Sub AsignarCategoriaAProductoToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AsignarCategoriaAProductoToolStripMenuItem.Click

asignar\_categoria\_producto.Show()

End Sub

Private Sub AjustarStockToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AjustarStockToolStripMenuItem.Click

ajustar\_stock.Show()

End Sub

Private Sub IngresarFacturaToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IngresarFacturaToolStripMenuItem.Click

ingreso\_factura.Show()

End Sub

Private Sub TerminarArriendoToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

arriendos.Show()

End Sub

Private Sub AcercaDeToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AcercaDeToolStripMenuItem.Click

MessageBox.Show(" Cualquier duda envie un Email a : 'sergio94mora@gmail.com; andresquezada121@gmail.com'")

End Sub

End Class