**Informe de proyecto**

**Sistema de gestión para comercios gastronómicos**

****

**Integrantes:**

* Larroca Agustin - 23910
* Vasquez Elmer - 23428

**Materia:**

* Laboratorio de computación II

**Profesores:**

* Ángel Simón
* Brian Lara

**Año:**

* 2020

**Introducción:**

El sistema de gestión propuesto está enfocado en poder agilizar el registro y control de movimientos de un local gastronómico. Para lograr un correcto uso de la aplicación, se dispone de un menú en el cual se podrá navegar libremente para poder acceder a diferentes opciones tales como registrar ventas totales de un cajero, monto recaudado hasta la fecha, usuarios activos y diferentes reportes.

**Objetivos:**

El objetivo de la aplicación es poder brindar al usuario una herramienta en la cual podrá ser capaz de registrar y visualizar diferentes movimientos que se generen en el local, detallando los datos obtenidos para una mayor precisión de análisis.

Uno de los mayores problemas que se enfrenta la gerencia de un local es la seguridad con la que se manejan los datos. En base a eso, nos orientamos a mejorar la seguridad al momento de interactuar con la aplicación, por lo que tendrá diferentes usos dependiendo de quién ingrese.

Los usuarios que se registren en el sistema tendrán un rol especifico con sus respectivos permisos, ya que no todos los usuarios podrán interactuar de la misma manera con la aplicación. Estos roles se dividirán en:

1. Gerente. ***Permisos absolutos***
2. Encargado de turno. ***Permisos limitados.***
3. Cajero. ***Permiso para acceder a registro de venta de Usuario.***

*Permiso absoluto: Acceso a todos los módulos.*

*Permiso limitado: Acceso a todos los módulos excepto a Retiro de efectivo*

Los datos necesarios para poder dar de alta a un usuario son:

* Nombre
* Apellido
* DNI
* Fecha de nacimiento
* Fecha de ingreso
* Rol

*Para un control interno de la aplicación, a medida que se vayan dando de alta cada usuario también tendrá un ID autonumérico.*

Cada vez que se quiera registrar un nuevo usuario deberán ser dados de alta por otro de mayor rango, una vez registrados en el sistema se les solicitará una contraseña única que deberán generar e ingresar cada vez que quieran acceder ya que esta pasará a guardarse como un atributo más del usuario.

*Existirá un usuario de soporte creado automaticamente por la aplicación con los mismos permisos de gerente y que solo debe ser utilizado para crear al primer usuario del sistema o en caso de que el gerente no pueda acceder al sistema.*

**Movimientos y gestión financiera:**

Se entiende que cada local posee una caja registradora con sus respectivas ventas, pero para establecer un segundo control podemos registrar las mismas en el sistema. Tendremos el módulo de movimientos donde se podrá registrar la entrada y salida de efectivo, las ventas que realicen los cajeros y acceso a la caja fuerte del local donde se almacena los montos netos recaudados.

La aplicación nos permitirá registrar las ventas de cada cajero obteniendo los siguientes datos:

**VENTA:**

* Usuario
* Productos vendidos
* Monto total

A modo de control, tendremos la posibilidad de saber cuánto efectivo recaudado disponemos en el local y como medida de seguridad, solamente el gerente del local podrá ser capaz de retirar ese monto.

**CAJA FUERTE:**

* Efectivo recaudado
* Retiro de efectivo

**Reportes:**

Una de las principales características de esta aplicación es la posibilidad de obtener reportes en los cuales podremos visualizar diferentes datos y movimientos realizados en la aplicación por seguridad. Los clasificaremos de la siguiente manera:

**VENTAS:**

* Ventas por fecha
* Ventas por usuario
* Ventas por producto
* Ventas por número de factura

**Aclaración:** Los datos que engloba este reporte serán **número de factura, usuario, monto total.**

**USUARIOS:**

* Activos
* Inactivos
* Bloqueados

**Aclaración:** Se toma como “Bloqueado” a los usuarios que ingresaron 3 veces erróneamente la contraseña.

Para que se pueda obtener un registro de la franja horaria en la que cada cajero empieza y termina de registrar ventas, automaticamente se tomará como horario de entrada el ingreso a la aplicación del empleado, así como también el horario de salida será el egreso. Al momento de salir de la aplicación, si el cajero realizo ventas la aplicación nos mostrará un reporte de las ventas totales que realizó.

**FICHAJES:**

* Fecha y usuario.

**Aclaración:** Este horario de fichaje podrá ser modificado por usuarios de nivel 1 y 2 en caso de tener que ajustar las horas trabajadas.

**Módulo administrativo:**

Dado que se le debe brindar al local una herramienta administradora que pueda asegurar la información recopilada hasta el momento, tendremos la opción de un módulo aparte en el cual podremos realizar backups de todos los datos generados para que el gerente pueda realizar una copia de seguridad cuando considere necesaria.

En el rubro gastronómico se fomenta el crecimiento de los empleados para lograr una rotación exitosa y un ambiente laboral cómodo, tendremos la posibilidad de cambiar el rol que se le asigno a un usuario originalmente. Solamente podrán acceder a este módulo los usuarios con rol de ‘Gerente’ o ‘Asistente’ ya que también podrán dar de alta o baja a otros usuarios.

Por último, para que cada cajero pueda registrar las ventas que realice el local podremos dar de alta o baja los productos que se encuentren a la venta.

**ADMINISTRACIÓN:**

* Alta de usuarios
* Baja de usuarios
* Modificación de rol
* Modificación de horario de fichaje
* Alta de productos
* Baja de productos
* Realizar copia de seguridad
* Restaurar copia de seguridad

**Anexo 1: Diagrama de clases**

|  |
| --- |
| **Usuario** |
| -ID: int  -Nombre: char[50]  -Apellido: char[50]  -DNI: int  -fechaDeNacimiento: Fecha  -fechaDeIngreso: Fecha  -rol: int  -contraseña: char[50]  -estado: bool |
| +ID: int  +SetfechaDeIngreso(): Fecha  +SetNombre(): char  +SetApellido(): char  +SetDNI(): int  +SetfechaDeNacimiento(): Fecha  +GetId(): int  +GetfechaDeIngreo(): Fecha  escribirDisco(pos:int): bool  leerDisco(pos:int): bool |

|  |
| --- |
| **Producto** |
| #precio: float  #descripcion: char[20] |
| +setPrecio(int): void  +setDescripcion(char \*):  +getPrecio(): int  +getDescripcion(): char \* |

|  |
| --- |
| **Fichaje** |
| -horarioDeIngreso: Horario  -horarioDeSalida: Horario  -fecha: Fecha  -idUsuario: int |
| +SetHorarioDeIngreso(): Horario  +SetHorarioDeSalida(): Horario  +GetHorarioDeIngreso(): Horario  +GetHorarioDeSalida(): Horario  +GetFecha(): Fecha  +GetIdUsuario (pos: int) |

|  |
| --- |
| **Fecha** |
| -dia: int  -mes: int  -Año: int |
| +setDia(): int  +setMes(): int  +setAño(): int  +getDia(): int  +getMes(): int  +getAño(): int |

|  |
| --- |
| **Venta** |
| -idUsuario: int  -fecha: Fecha  -numeroDeFactura: int  -productosVendidos: productoVendido[10]  -formaDePago: bool |
| setProductosVendidos(): productoVendido  setFormaDePago(): bool  GetProductosVendidos(): productoVendido  GetFormaDePago(): bool  escribirDisco(pos:int): bool  leerDisco(pos:int): bool |

|  |
| --- |
| **Horario** |
| -horas: int  -minutos: int |
| +setHoras(): void  +setMinutos(): void  +getHoras(): int  +getMinutos(): int |

|  |
| --- |
| **ProductoVendido** |
| -cantidad: int |
| +setCantidad(cant: int): void  +getCantidad(): int |

|  |
| --- |
| **CajaFuerte** |
| -saldo: int |
| +setSaldo(valor int): void  +getSaldo(): int  +retirarEfectivo(valor: int): void |