

Análisis de Lenguajes de Programación



Trabajo Práctico Final

Administrador de Granja

1 Introducción

El programa está pensado para pequeñas o medianas granjas, kioskos o emprendimientos que tengan stock de ciertos productos y realicen ventas.

Está desarrollado y probado desde Ubuntu, pero debería ser compatible para otras distribuciones y plataformas. Utiliza stack para manejar el proyecto.

1.1 Secciones y uso

- Productos:

Acá figuran todos los productos disponibles a la venta con su información la cual se puede editar: nombre, proveedor, código, stock y precio. Además permite agregar y eliminar productos.

- Vender:

Permite realizar ventas de los productos cargados, el funcionamiento es el siguiente:

Para agregar productos a la venta ingresar la cantidad de unidades a vender y luego la tecla más.

Para retirar productos de la venta ingresar la cantidad de unidades a borrar y luego la tecla menos.

Una vez establecida la cantidad de unidades a agregar o retirar, que por defecto es una a agregar, se ingresa el código del producto y se presiona enter.

Para completar la venta se presiona enter.

- Ventas:

Muestra el historial de ventas ordenadas cronológicamente y permite eliminar las mismas.

1.2 Instalación

Una vez clonado el repositorio se deben instalar las siguientes dependencias:

```
sudo apt-get install sqlite3
sudo apt-get install libsqlite3-dev
stack install HDBC
stack install HDBC-sqlite3
sudo apt-get install libgirepository1.0-dev libwebkit2gtk-4.0-dev libgtksourceview-3.0-dev
stack install gi-gtk
```

⚠ Una de las dependencias (haskell-gi) tiene un bug con GHC 8.2.x por lo que hay que evitar dicha versión.

Se compila y ejecuta con el siguiente comando:

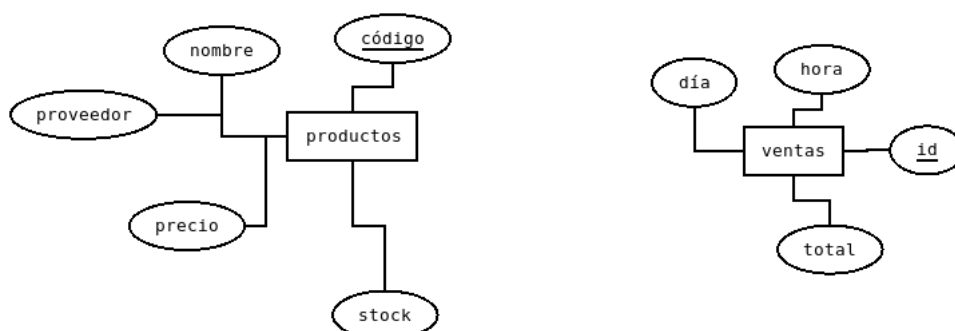
```
stack build && stack exec Administrador-Granja-exe
```

2 Base de datos

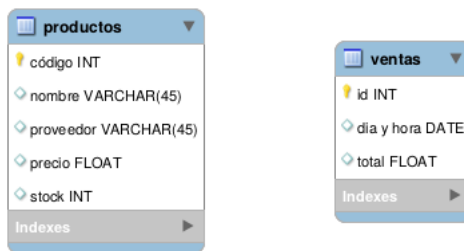
Para llevar un registro persistente de la información se utiliza una base de datos. Particularmente se utiliza SQLite3, que es una implementación de base de datos SQL en C, junto con el paquete HDBC (Haskell Database Connectivity) que permite utilizar SQLite3 desde Haskell.

La información que se aloja en la base de datos son los productos y el historial de las ventas realizadas.

El diagrama entidad/relación utilizado es el siguiente:



Y su pasaje a tablas el siguiente:



3 Interfaz gráfica

Se utilizó GTK3 para implementar la interfaz gráfica, que es una biblioteca implementada en C para este propósito. Además se utilizó el paquete `haskell-gi` que permite utilizar GTK3 desde Haskell.

Para una pequeña y rápida introducción a GTK cito el siguiente fragmento del manual de referencia:

GTK+ is a widget toolkit. Each user interface created by GTK+ consists of widgets. This is implemented in C using GObject, an object-oriented framework for C. Widgets are organized in a hierarchy. The window widget is the main container. The user interface is then built by adding buttons, drop-down menus, input fields, and other widgets to the window. If you are creating complex user interfaces it is recommended to use GtkBuilder and its GTK-specific markup description language, instead of assembling the interface manually. You can also use a visual user interface editor, like Glade.

GTK+ is event-driven. The toolkit listens for events such as a click on a button, and passes the event to your application.

3.1 Glade

Para facilitar la implementación de la interfaz gráfica se utilizó Glade, que es un programa que cumple este propósito. Permite diseñar la interfaz gráfica (crear y agregar ventanas, widgets, etc) de una manera sencilla y visual. Su output es un archivo XML que en nuestro caso es `gui.glade` que se utiliza para crear un GtkBuilder que nos facilita el proceso. El archivo `gui.glade` está pensado para ser visto y editado justamente desde Glade.

4 Organización de los archivos

```
Administrador-Granja
├── assets:  archivos relevantes al proyecto que no son código (imágenes, informe, etc).
├── app:    aloja el módulo Main.
├── Base de datos:  aloja la base de datos.
├── src:    aloja todo el código excepto el módulo Main.
│   ├── GUI: aloja los módulos relacionados con la GUI.
│   │   ├── GUI.hs:  exporta la función gui que es la que inicializa la GUI.
│   │   ├── Productos.hs:  módulo auxiliar para la sección Productos.
│   │   ├── Vender.hs:  módulo auxiliar para la sección Vender.
│   │   ├── Ventas.hs:  módulo auxiliar para la sección Ventas.
│   │   └── Misc.hs:  exporta funciones para mostrar mensajes.
│   ├── BibliotecaBD.hs:  exporta funciones útiles para manejar la base de datos.
│   ├── Estructuras.hs:  describe la estructura Producto utilizada.
│   └── Inicializador.hs:  exporta la función iniciar que crea las tablas si no están creadas.
└── test:  no lo utilizamos (creado por stack).
```

5 Decisiones de diseño

Una decisión de diseño a destacar es en la diferencia del cálculo del total para las secciones Vender y Ventas.

Para Vender decidí que cada vez que se modifica la lista de los productos por vender se calcula el total nuevamente iterando toda la lista. Esto se hizo así por la simplicidad de implementarlo y considerando que la lista de productos a vender nunca va a ser realmente grande.

Por otro lado, en la sección Ventas, considere que calcular el total recorriendo toda la lista por cada venta agregada o removida podría ser un problema si el historial de ventas es considerable. Así que, por ejemplo, para agregar una venta se lee el total, se le suma el importe de la nueva venta y se asigna este valor como nuevo total. Al eliminar una venta se aplica el mismo procedimiento pero se le resta al total el importe de la venta eliminada.

6 Posibles mejoras

- Asignar teclas para seleccionar cada Stack (o sección) (por ejemplo F1, F2 y F3).
- Mejorar el historial de ventas: que quede registrado que productos se vendieron.
- Mejorar la visualización de las ventas: por ejemplo poder ver las ventas por día/mes/año.
- Stock con cantidades reales en vez de enteras.
- Cuando se concreta una venta, para actualizar el stock de los productos en la store de productos la misma se recorre una a una por cada producto. Podría ser un proceso lento si hay muchos productos. Evitar esto sería una mejora.

7 Links útiles

- [stack \(instalación\)](#)
- [HDBC \(documentación\)](#)
- [GTK3 \(manual de referencia\)](#)
- [gi-gtk \(documentación\)](#)
- [Glade \(sitio oficial\)](#)