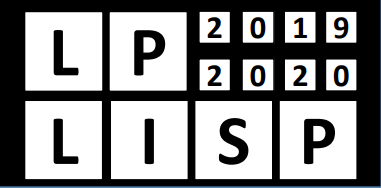
Imagen que contiene señal

Descripción generada automáticamente

Práctica no presencial LISP

21721 - Lenguajes de Programación



Nadal Llabrés Belmar (Grupo 01X01)

Andreu López Cortés (Grupo 01X02)

15/06/2020

Índice

[Introducción 3](#_Toc41810144)

[Explicación del enunciado 3](#_Toc41810145)

[Diagrama de flujo 4](#_Toc41810146)

[Requisitos y material adjuntado 5](#_Toc41810147)

[Diseño descendente 5](#_Toc41810148)

[Funciones principales 6](#_Toc41810149)

[Manual de usuario 7](#_Toc41810150)

[Carga del código 7](#_Toc41810151)

[Manejo del programa 8](#_Toc41810152)

[Pruebas y test 13](#_Toc41810153)

[Pruebas estáticas 13](#_Toc41810154)

[Pruebas dinámicas 13](#_Toc41810155)

[Conclusión 14](#_Toc41810156)

# Introducción

Esta es la documentación correspondiente a la práctica de la primera parte de la asignatura 21721 – Lenguajes de programación de la Universidad de las Islas Baleares. Es una práctica desarrollada con el lenguaje LISP y ejecutada mediante el intérprete XLISP-PLUS versión 3.04.

Para escribir el código se ha utilizado el IDE Visual Studio Code junto con la extensión del lenguaje LISP.

# Explicación del enunciado

Lo que se pide hacer es una tienda de productos en la que el usuario pueda elegir los deseados e ir acumulándolos en un carrito de compra de mientras va visualizando el total en cada momento.

La tienda dispone de veinte productos enumerados el 0 al 19 los cuáles son visibles en todo momento.

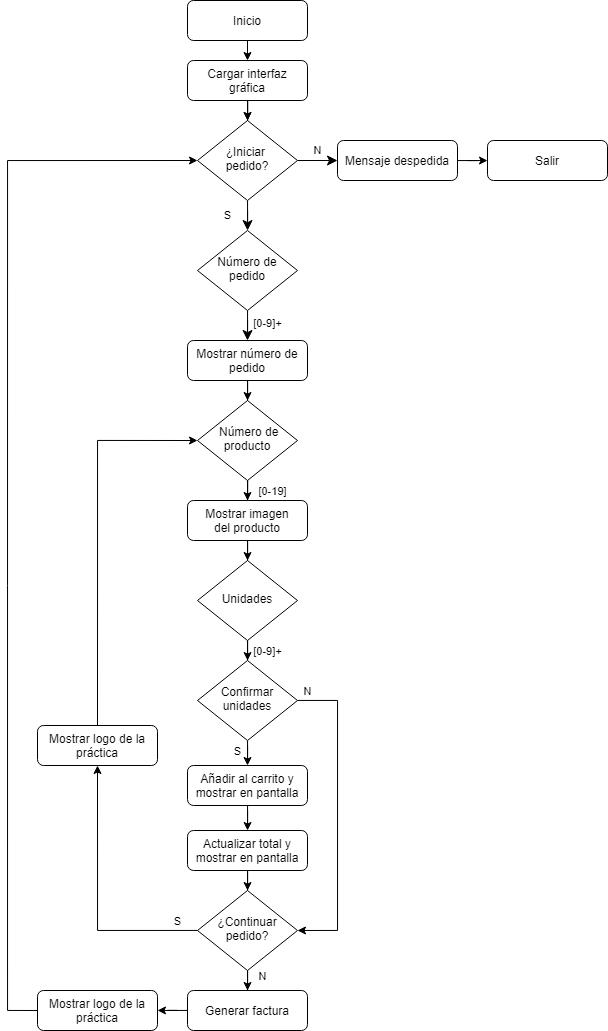
El objetivo es que el usuario introduzca un número de pedido que será en el cuál irá almacenando los productos. Después de introducir el número irá introduciendo los productos, los cuales podrá visualizar en una ventana de imagen y elegir el número de unidades de cada uno.

Una vez el usuario haya terminado el producto, el programa deberá generar un archivo llamado pedidoX.txt (donde X es el número de pedido) donde se pueda ver en la primera línea el número de pedido, el listado de productos, las unidades de cada uno, el precio de cada producto por las unidades y el total.

**NOTA:** Todas las imágenes que se utilicen tienen que estar en formato .bmp de Microsoft Windows.

## Diagrama de flujo

Aquí se puede ver el flujo que tiene que seguir el programa.



## Requisitos y material adjuntado

Los requisitos principales que se piden son:

* Se debe entregar un fichero con extensión .lsp en el que habrá una función sin argumentos llamada “inicio” que será la que empezará toda la ejecución.
* El código tiene que estar comentado y estructurado.
* Entrega de todos los ficheros necesarios para el funcionamiento.

Como material adjunto por parte del profesorado se aporta:

* Imagen del logo de la práctica.
* Imágenes de cada una de las letras del abecedario y números del 0 al 9.

# Diseño descendente

El código del programa ha sido estructurado de forma que sea entendible para el programador, se consiga reducir el número de líneas y se obtenga la máxima claridad posible.

Veamos las partes principales del código:

* Definición de constantes.
* Introducción de productos.
* Funciones del programa.
  + Función inicio.
* Funciones de apoyo.

A continuación, una explicación de cada punto:

**Definición de constantes**

Para empezar, en las primeras líneas se pueden ver una serie de constantes definidas que permiten que el código sea más legible y realizar algunos cambios fácilmente.

**Introducción de productos**

Los productos han sido introducidos en forma de lista de listas de productos. Eso significa que hay una lista de 20 elementos que contiene todos los productos, pero a su vez, cada producto es una lista de 2 elementos (nombre del producto y precio).

producto = (“Nombre” precio)

Lista de productos = (producto0 producto1 producto2…)

**Funciones del programa**

Aquí se encuentran todas las funciones que gestionan toda la lógica del programa. Ejemplo: Menu, generateBill, addToCart, updateTotal, printCart, printProducts, etc.

**Funciones de apoyo**

Las funciones de este bloque son funciones que trabajan a un nivel más bajo. Por ejemplo, las encargadas de dibujar los gráficos píxel a píxel o mover el *prompt*. Muchas de ellas han sido reutilizadas de los ejemplos de aula digital proporcionadas por el profesor.

# Funciones principales

A continuación, se enumerarán y explicarán las funciones más relevantes de las que consta el programa:

* Inicio
* Menu
* generateBill
* printCart
* printProducts
* initGuiElements
* printWord
* printImage
* fillAreaColor

**Inicio**

Es la función a la que hay que llamar para lanzar la ejecución del programa. Forma parte de los requisitos de la práctica.

**Menu**

Contiene el bucle del menú de opciones que permitirán al usuario interactuar con la interfaz y conformar los pedidos.

**generateBill**

Genera una factura de los productos seleccionados por el usuario en formato de archivo de texto.

**printCart**

Imprime los productos que están en ese momento en el carrito en la parte de la ventana reservada para ello.

**printProducts**

Imprime todos los productos de los que dispone la tienda en la zona de la ventana asignada.

**initGuiElements**

Inicia los elementos gráficos de la ventana. Por ejemplo:

* Cabecera
* Zona de productos
* Zona número de pedido
* Zona de carrito
* Precio total

**printWord**

Recibe un string por parámetro y una posición x, y e imprime mediante las imágenes de caracteres el texto.

**printImage**

Imprime la imagen indicada por parámetro en la posición x, y con el tamaño también indicado por parámetro.

**fillAreaColor**

Rellena un área que se indica por parámetro mediante coordenadas con el color pasado también por parámetro en formato RGB.

# Manual de usuario

El programa tiene una interfaz gráfica muy intuitiva y fácil de utilizar. No obstante, a continuación, se explicará con detalle:

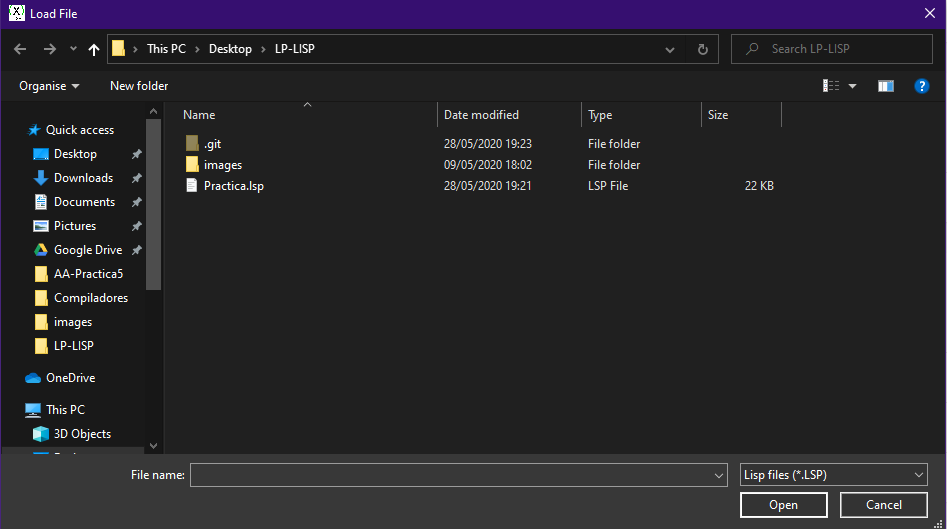
## Carga del código

Al abrir el intérprete hay que cargar el código del programa. Para ello hay que hacer clic en File 🡪 Open/load…

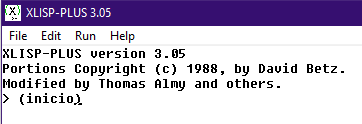
Captura de pantalla de un celular

Descripción generada automáticamente

A continuación, hay que navegar hasta el directorio del proyecto y abrir el fichero “Practica.lsp”.



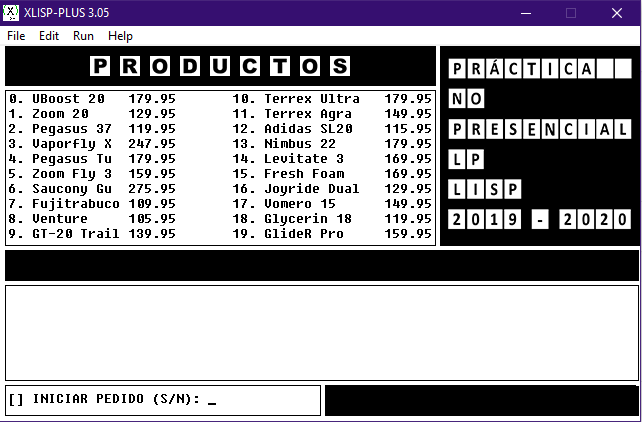
Una vez abierto, llamara la función de inicio de la siguiente manera:



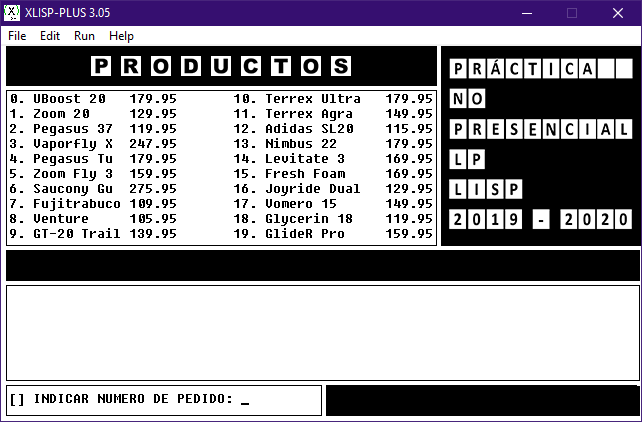
Automáticamente se iniciará el programa y se podrá utilizar.

## Manejo del programa

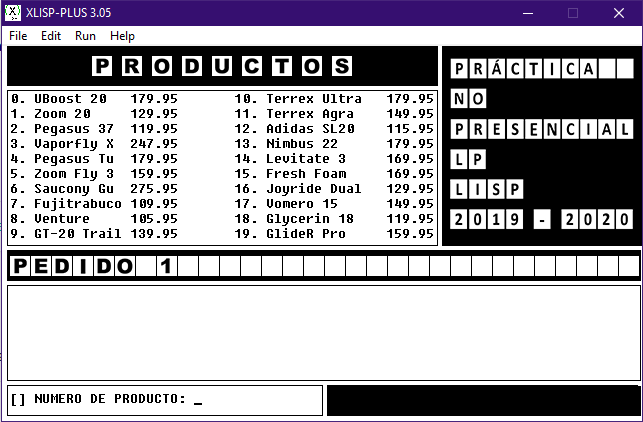
Por defecto la interfaz es la que aparece en la siguiente imagen. Empezamos nuevo pedido introduciendo “s” y a continuación enter para confirmar.



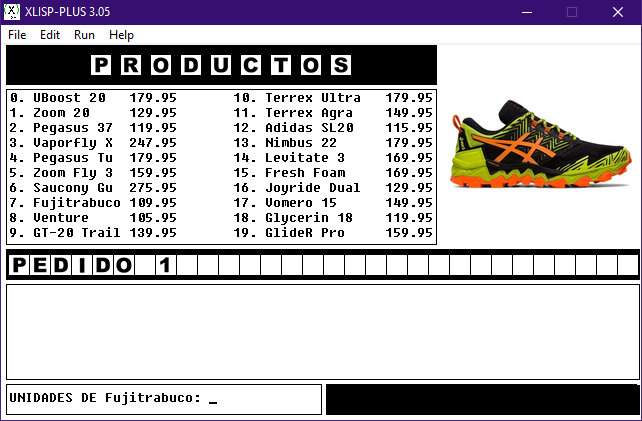
Indicamos un número de pedido que será con el que se guardará la factura.



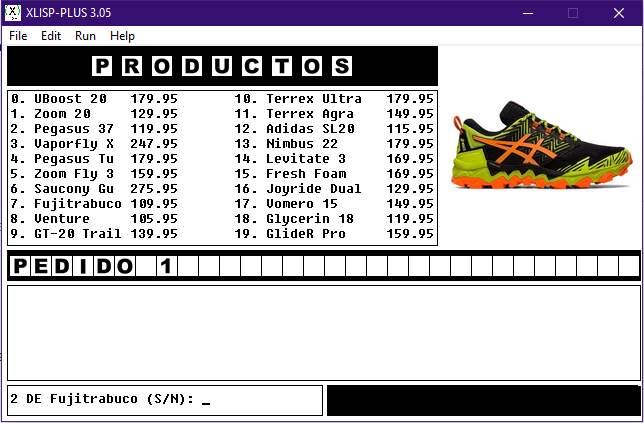
El siguiente paso es ir indicando los productos que deseamos a través de su identificador.



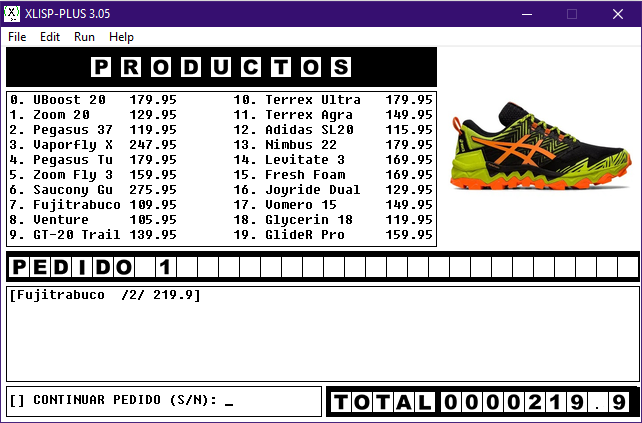
Al indicar el producto nos aparecerá una imagen i se podrá indicar el número de unidades.



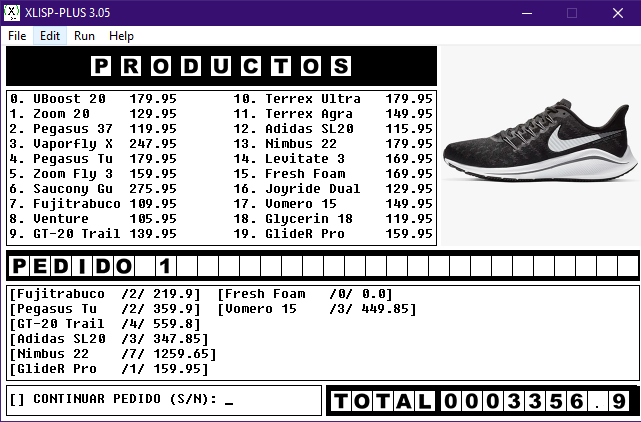
Confirmamos el número de unidades.



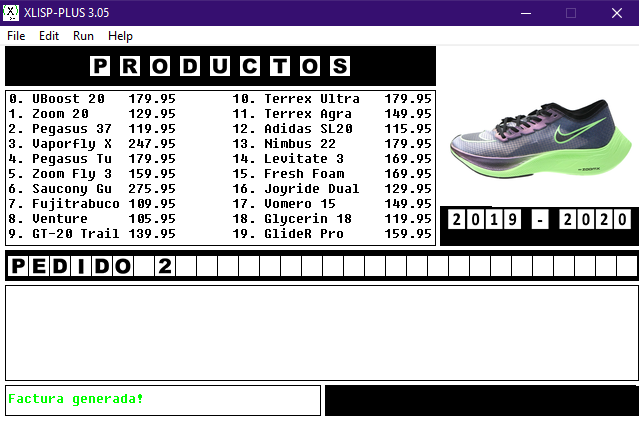
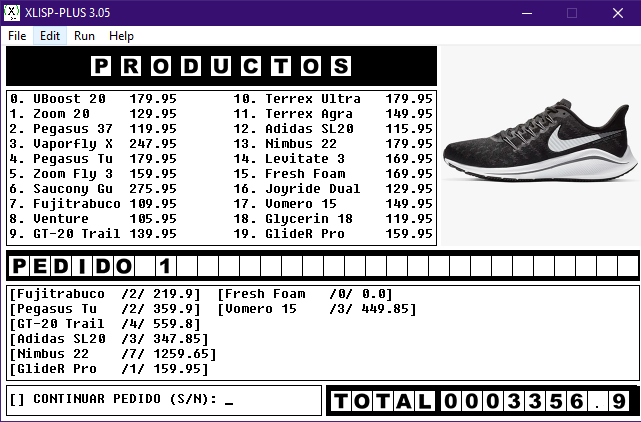
Vemos que se ha añadido a la lista de productos seleccionados. Podemos continuar comprando productos o parar.



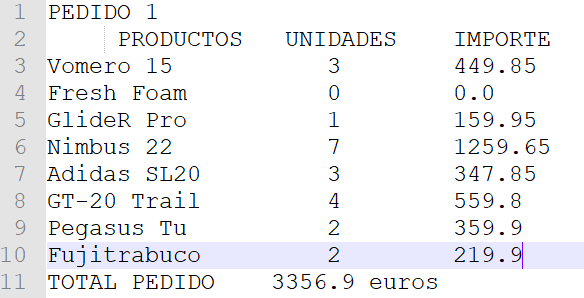
Vamos añadiendo todos los que se deseen y de mientras se puede ir viendo el total en la parte inferior derecha.



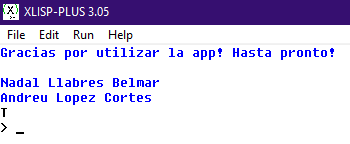
Al terminar de añadir productos, indicamos que no queremos continuar el pedido y automáticamente se generará la factura.



Podemos consultar la factura y allí veremos todos los productos con sus unidades, importe y total del pedido.



Finalmente, la interfaz volverá a su estado inicial y se podrá crear un pedido nuevo o salir de la aplicación.



# Pruebas y test

Para verificar que el programa cumple con los requisitos establecidos por el enunciado de la práctica se han realizado varias pruebas. Las pruebas realizadas se dividen en:

## Pruebas estáticas

Estas pruebas se realizan sin ejecutar el código del programa. Los tests realizados son:

* + Revisión de comentarios en todas las funciones del programa.
  + Cabecera del programa donde se indica los participantes, curso académico y asignatura.
  + Mirar que no haya quedado ningún trozo de código de pruebas.
  + Verificar que todos los nombres de las funciones y variables del programa siguen el mismo patrón.
  + Verificar que las imágenes de los productos son de tamaño 200x200, extensión “.bmp”, profundidad de 24 bits y formato Microsoft Windows.
  + Comprobar la documentación contiene todos los apartados necesarios y las explicaciones son claras.
  + Revisar requisitos del enunciado y comprobar que se cumplen.

## Pruebas dinámicas

Al contrario que las anteriores, las dinámicas son las pruebas que requieren la ejecución de la aplicación.

* Visualización correcta de todas las imágenes de los productos.
* Realización de dos pedidos en la misma ejecución para ver que se reinicia el número de pedido, el precio total y el carrito.
* Comprobar que se carga el logo de la práctica cuando no hay ningún pedido seleccionado.
* Revisión de la factura. Incluye todos los productos, nombre de pedido, total y el texto está correctamente formateado.
* Probar todas las casuísticas posibles del diagrama de flujo.
* Verificar que se cumplen todos los requisitos del enunciado.

# Conclusión

Al principio parecía una práctica bastante complicada porque no conocíamos el lenguaje de programación LISP y, además, es bastante diferente a lo que estamos acostumbrados (Java, Python, etc). Finalmente, se ha podido ver que no ha sido tan complicado ya que el funcionamiento del lenguaje se aprende rápido e incluso ha resultado bastante divertido y motivador (algo que con este tipo de lenguajes no parece que tenga que ocurrir con tanta facilidad).

Ha sido interesante ver como se han manejado tres aspectos importantes como son la lógica del programa, dibujado de gráficos y escritura de ficheros.

La dificultad y extensión ha sido la correcta y esperada.