

# PLAN DE TRABAJO DEL ESTUDIANTE

## 1. INFORMACIÓN GENERAL

Apellidos y Nombres: Sotelo Cardenas Ana Cecilia ID: 001328493

Dirección Zonal/CFP: Ica - Ayacucho

Carrera: Ig de software con IA Semestre: 4º

Curso/ Mód. Formativo: ALGORITMIA PARA LA INTELIGENCIA ARTIFICIAL

Tema del Trabajo: Creación de algoritmos aplicados a linteligencia artificial

## 2. PLANIFICACIÓN DEL TRABAJO

Nº	ACTIVIDADES/ ENTREGABLES	CRONOGRAMA/ FECHA DE ENTREGA									
1	trabajo final										
2	entregar del cuaderno										
3											

### 3. PREGUNTAS GUIA

Durante la investigación de estudio, debes obtener las respuestas a las siguientes interrogantes:

Nº	PREGUNTAS
1	¿Cuáles son los medios de expresión de un algoritmo en inteligencia artificial?
2	¿Cómo se desarrolla y analiza los algoritmos en IA?
3	¿Describe en qué consiste la algoritmia aplicada a la IA?
4	¿Cuáles son técnicas de Diseño de algoritmos IA?5. ¿Describe en qué consiste aprendizaje evolutivo?
5	

**TRABAJO FINAL DEL CURSO**
**HOJA DE RESPUESTAS A LAS PREGUNTAS GUÍA**

1.	¿Cuáles son los medios de expresión de un algoritmo en inteligencia artificial?
Los métodos de expresión son los lenguajes de programación, un algoritmo en inteligencia artificial, se implementa utilizando lenguaje de programación que es adecuado para la tarea que se está abordando.	
2.	¿Cómo se desarrolla y analiza los algoritmos en IA?
Esto se puede resumir en los siguientes pasos: 1. Definir el problema, 2. Recopilar datos, 3. Preprocesar los datos, 4. Seleccionar un modelo de IA, 5. Entrenar el modelo, 6. Validar y evaluar el modelo, 7. Ajustar y mejorar modelo:	
3.	¿Describe en qué consiste la algoritmia aplicada a la IA?
Se referencia al proceso de diseñar y desarrollar que permitan al sistema, hacer llevar a cabo ciertas tareas específicas	
4.	¿Cuáles son técnicas de Diseño de algoritmos IA?
Las más comunes son: Redes neuronales, 2. Árboles de decisión, 3. Máquinas de vectores de soporte (SVM): 4. Algoritmos genéticos: 5. Agrupamiento (Clustering), 6. Aprendizaje por refuerzo:	
5.	¿Describe en qué consiste aprendizaje evolutivo?
es una técnica de aprendizaje automático que se inspira en la teoría de la evolución de Darwin. En lugar de utilizar un algoritmo de aprendizaje convencional, el aprendizaje evolutivo utiliza algoritmos genéticos para generar soluciones y mejorarlas a lo largo del tiempo.	

## TRABAJO FINAL DEL CURSO

## HOJA DE PLANIFICACIÓN

## PROCESO DE EJECUCIÓN

[illegible]

**INSTRUCCIONES:** debes ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. No olvides los aspectos de calidad, medio ambiente y SHI.

## TRABAJO FINAL DEL CURSO

### DIBUJO / ESQUEMA/ DIAGRAMA

```

tornillo_extra.py 9+ X
1 import calendar
2 import datetime
3 import time
4 from tkinter import *
5 from tkinter import messagebox as MessageBox
6
7 # Hora en tiempo real
8 def hora():
9
10     class App(Frame):
11         def __init__(self, master=None):
12             Frame.__init__(self, master)
13             self.master = master
14             self.label = Label(text="", font=("Calibri", 20))
15             self.label.place(x=490, y=10)
16             self.update_clock()
17             self.config(bg= "FFFFFF")
18
19         def update_clock(self):
20             now = time.strftime("%H:%M:%S")
21             self.label.configure(text=now)
22             self.after(1000, self.update_clock)
23
24
25
26 app=App(root)
27 root.wm_title("Ferreteria")
28 app.config(bg= "FFFFFF")
29
30 root.resizable(width=False, height=False)
31 root.after(1000, app.update_clock)
32
33
34 # Calendario en tiempo real
35 def calendario():
36
37
38 # Obtenemos los valores del año y mes a mostrar
39 año = datetime.date.today().year
40 mes = datetime.date.today().month
41
42 def escribirCalendario(año,mes):
43     # Asignamos el año y mes al calendario
44     str1 = calendar.month(año,mes)
45
46     label1.configure(text=str1)
47

```

En esta primera parte, desde la línea 1 a la línea 5 se importaran las librerías que necesitaremos para la realización de nuestro proyecto.

De la línea 8 a la línea 22 se define al función reloj, los cuales señala los atributos, como la hora, minuto y segundo. También se le incluye un color de fondo y el estilo de fuente. Luego de la línea 26 a la 31, se muestra el código que permitirá mostrar un titulo en la cabecera de la ventana. Y desde la línea 38 a la línea 46 vemos la definicion de lafuncion calendario, con los atributos días, meses y años.

## TRABAJO FINAL DEL CURSO

```

41
42 def escribirCalendario(año,mes):
43     # Asignamos el año y mes al calendario
44     str1 = calendar.month(año,mes)
45
46     label1.configure(text=str1)
47
48 def mesAnterior():
49     global month,year
50     mes-=1
51     if mes==0:
52         mes=12
53         año-=1
54
55     escribirCalendario(año,mes)
56
57 def mesSiguiente():
58     global month,year
59     mes+=1
60     if mes==13:
61         mes=1
62         año+=1
63
64     escribirCalendario(año,mes)
65
66 root = Tk()
67 root.title("Calendario")
68 root.resizable(False,False)
69
70 # Lo posicionamos en un label
71 label1 =Label(root, text="", font=('courier', 14, 'bold'), bg='white', justify=LEFT)
72 label1.grid(row=1,column=1)
73
74 # ponemos los botones dentro un Frame
75 frame=Frame(root,bd=5)
76
77
78 anterior = Button(frame,text="Anterior", command=mesAnterior)
79 anterior.grid(row=1, column=1, sticky=W)
80 siguiente =Button(frame,text="Siguiente", command=mesSiguiente)
81 siguiente.grid(row=1, column=2)
82 frame.grid(row=2,column=1)
83
84 escribirCalendario(año,mes)
85
86
87

```

Después de crear las variables donde se almacene el valor, creamos los botones para el mes anterior y el siguiente, los ubicamos mediante la posición de columnas y filas.

## TRABAJO FINAL DEL CURSO

```

File Edit Selection View Go Run Terminal Help
tornillo_extra.py 9+ X
def administrador():
    """De momento esta funcion está vacio """
    root = Tk() #root = nombre de mi raíz
    root.title("Ferretería El tornillo feliz") # nombre del titulo de mi ventana
    root.config(bg= "#BDDCE1") # background o fondo de mi ventana
    #root.iconbitmap("imagen.ico") # imagen con el icono de mi ventana
    root.resizable(False,False) #Evita que se expanda mi ventana

    miFrame = Frame(root)# primer frame en la funcion administrador
    miFrame.pack() #empaquetando
    miFrame.config(bg= "#A4D8D4")#fondo de mi primer frame
    miFrame.config(cursor = "pirate") #miniatura del raton (pirata)

    #Insertamos los widgets
    #Label titulo
    lDni = Label(miFrame, text='Registrar', font=("calibri", 25),bg= "#A4D8D4" )
    lDni.grid(row=5, column=4, pady=15, padx=15)

    #-----
    #Label y entry de DNI -----
    obtenerDni=StringVar()
    lDni = Label(miFrame, text='Usuario',bg = "#FFEE00")
    lDni.grid(row=5, column=1, sticky='e', pady=5, padx=5)
    tDni = Entry(miFrame,textvariable=obtenerDni)
    tDni.grid(row=5, column=2, pady=5, padx=5)

    #Label y entry Apellido -----
    obtenerApellido=StringVar()
    lApellido = Label(miFrame, text='Contraseña',bg = "#FFEE00")
    lApellido.grid(row=7, column=1, sticky='e', pady=5, padx=5)
    tApellido = Entry(miFrame,textvariable=obtenerApellido)
    tApellido.grid(row=7, column=2, pady=5, padx=5)

    #Label y entry Nombre -----
    obtenerNombre=StringVar()
    lNombre = Label(miFrame, text='País',bg = "#FFEE00")
    lNombre.grid(row=9, column=1, sticky='e', pady=5, padx=5)
    tNombre = Entry(miFrame,textvariable=obtenerNombre)
    tNombre.grid(row=9, column=2, pady=5, padx=5)

    #Label y entry Dirección -----
    obtenerDir=StringVar()
    lDireccion = Label(miFrame, text='Empresa',bg = "#FFEE00")
    lDireccion.grid(row=11, column=1, sticky='e', pady=10, padx=10)
    tDireccion = Entry(miFrame,textvariable=obtenerDir)
    tDireccion.grid(row=11, column=2, columnspan=3, sticky='we',pady=10, padx=10)

```

Definimos la función administrador, con sus atributos, como el color de fondo o la función del ratón pirata.

Luego las otras cajas de texto que se van a codificar de la misma manera, con un título, la ubicación en columnas y filas, y el color de fondo

## TRABAJO FINAL DEL CURSO

```
File Edit Selection View Go Run Terminal Help
tornillo_extra.py > ...
151 def add():
152
153     """De momento vacio"""
154     #Falta ayadir funciones par el usuario final
155
156     root = Tk() #root = nombre de mi raíz
157
158     root.title('Informacion') # nombre del titulo de mi ventana
159     root.config(bg= "#B0C4DE") # background o fondo de mi ventana
160     #root.iconbitmap('imagen/ico') # imagen con el icono de mi ventana
161     root.resizable(False,False) #Evita que se expanda mi ventana
162
163
164     miFrame = Frame(root)# primer frame en la funcion administrador
165     miFrame.pack() #empaquetando
166     miFrame.config(bg= "#A4D8D4")#fondo de mi primer frame
167     miFrame.config(cursor= "pirate") #miniatura del raton (pirata)
168
169     #Insertamos los widgets
170     #label titulo
171     lDni = Label(miFrame, text='Programa en mantenimiento', font=("calibri", 40),bg= "#A4D8D4" )
172     lDni.grid(row=0, column=3, pady=15, padx=15)
173
174     lDni1 = Label(miFrame, text='''Por ahora no hay nada que añadir
175     disculpe las molestias, Esperamos su comprension
176     ''', font=("calibri", 25),bg= "#FFEEB8" )
177     lDni1.grid(row=3, column=3, pady=15, padx=15)
178
179     lDni1 = Label(miFrame, text='''El programador está de vacaciones
180     sin fecha definida.
181     ''', font=("calibri", 15),bg= "#A4D8D4" )
182     lDni1.grid(row=5, column=3, pady=15, padx=15)
183
184
185
186
187
188 def home():
189     """De momento esta funcion está vacio """
190     #Falta ayadir funciones par el usuario final
191
192
193 def info():
194     """De momento esta funcion está vacio """
195     #Falta ayadir funciones par el usuario final
196     """De momento esta funcion está vacio """
197     root = Tk() #root = nombre de mi raíz
```

Definimos la función add, será la función mediante la cual codificaremos la ubicación, estilo de texto, tamaño y color de fondo de las opciones del menú que se despliegue de la pestaña de arriba



## TRABAJO FINAL DEL CURSO

```

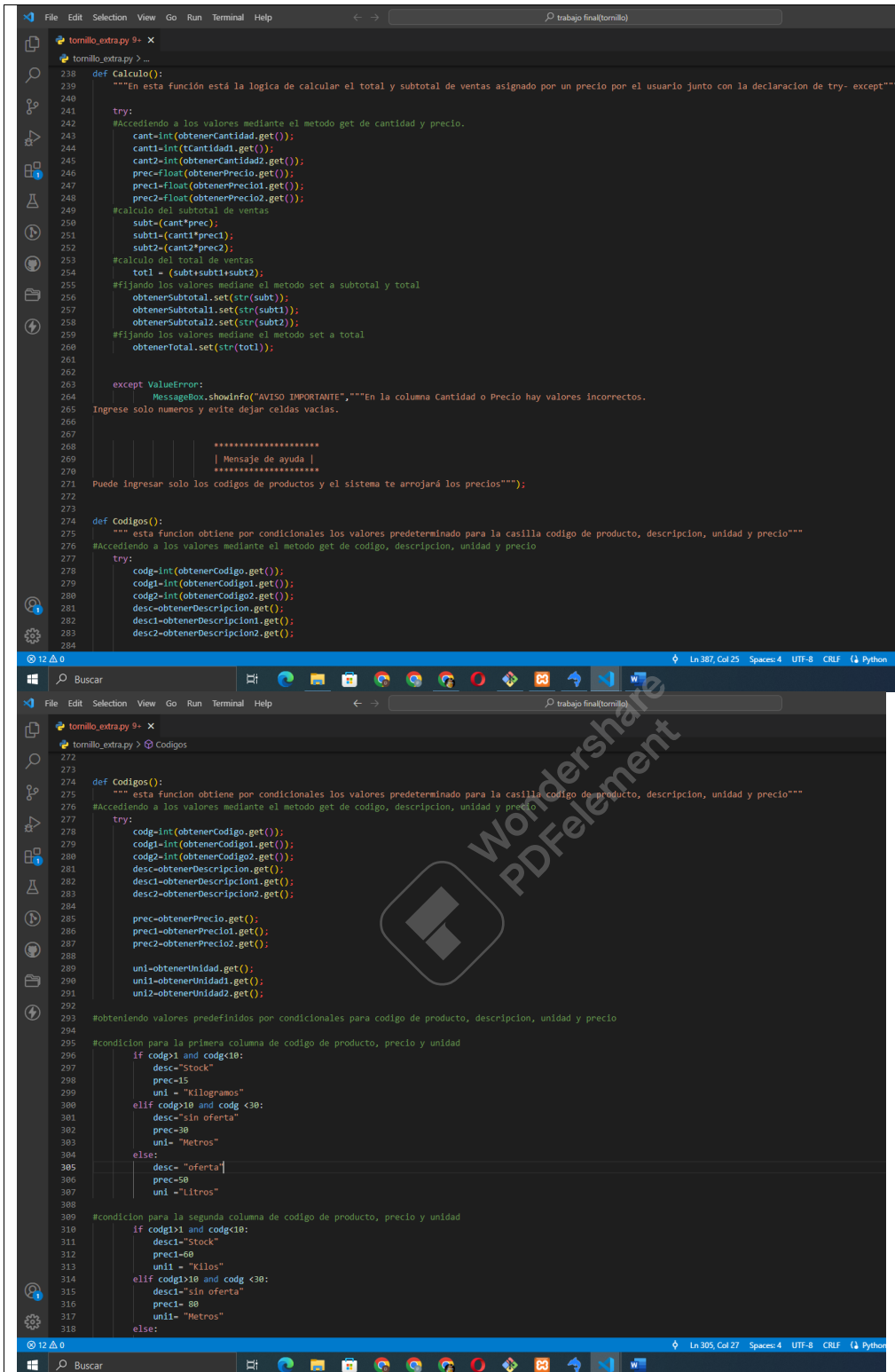
188 def home():
189     """De momento esta funcion está vacío"""
190     #Falta ayadir funciones par el usuario final
191
192
193 def info():
194     """De momento esta funcion está vacío"""
195     #Falta ayadir funciones par el usuario final
196     """De momento esta funcion está vacío"""
197     root = Tk() #root = nombre de mi raíz
198
199     root.title('Informacion') # nombre del título de mi ventana
200     root.config(bg= "#D0CE1") # background o fondo de mi ventana
201     #root.iconbitmap('imagen.ico') # imagen con el icono de mi ventana
202     root.resizable(False,False) #Evita que se expanda mi ventana
203
204
205     miFrame = Frame(root)# primer frame en la funcion administrador
206     miFrame.pack() #respauntando
207     miFrame.config(bg= "#A4D8D4")#Fondo de mi primer frame
208     miFrame.config(cursor = "pirate") #miniatura del raton (pirata)
209
210 #insertamos los widgets
211 #label titulo
212 lDn1 = Label(miFrame, text="PROYECTO DE:", font= ("calibri", 40),bg= "#A4D8D4" )
213 lDn1.grid(row=0, column=3, pady=15, padx=15)
214
215 lDn11 = Label(miFrame, text= "'LUKAS ATUNCAR VALERIO"
216             "'", font= ("calibri", 25),bg= "#A4D8D4" )
217 lDn11.grid(row=3, column=3, pady=15, padx=15)
218
219 lDn11 = Label(miFrame, text= "'Estudiante de la Carrera de Ingeniera de Software con Inteligencia Artificial"
220             "Cuarto Semestre'", font= ("calibri", 18),bg= "#A4D8D4" )
221 lDn11.grid(row=4, column=3, pady=15, padx=15)
222
223 lDn11 = Label(miFrame, text= "SENATI", font= ("calibri", 18),bg= "#A4D8D4" )
224 lDn11.grid(row=5, column=3, pady=15, padx=15)
225
226 lDn11 = Label(miFrame, text= "Programa de uso libre para aprendizaje"
227             "cualquier uso indebido tu cuenta será hackeada :)"
228
229
230 Estas advertido.
231 Todos los derechos Reservados
232             "'", font= ("calibri", 13),bg= "#A4D8D4" )
233 lDn11.grid(row=7, column=3, pady=15, padx=15)
234

```

Definimos la función info, que nos ayudará a poder visualizar la sección “información” del menú de arriba.

En la siguiente parte se programa toda la información que se verá en esta sección, ubicados en filas y columnas, estilos y tamaño de texto con el color de fondo.

## TRABAJO FINAL DEL CURSO



```

238 def Calculo():
239     """En esta función está la logica de calcular el total y subtotal de ventas asignado por un precio por el usuario junto con la declaracion de try- except"""
240
241     try:
242         #Accediendo a los valores mediante el metodo get de cantidad y precio.
243         cant=int(obtenerCantidad.get());
244         cant1=int(tcantidad1.get());
245         cant2=int(obtenerCantidad2.get());
246         prec=float(obtenerPrecio.get());
247         prec1=float(obtenerPrecio1.get());
248         prec2=float(obtenerPrecio2.get());
249         #calculo del subtotal de ventas
250         sub=(cant*prec);
251         sub1=(cant1*prec1);
252         sub2=(cant2*prec2);
253         #calculo del total de ventas
254         tot1 = (sub+sub1+sub2);
255         #fijando los valores mediante el metodo set a subtotal y total
256         obtenerSubtotal.set(str(sub));
257         obtenerSubtotal1.set(str(sub1));
258         obtenerSubtotal2.set(str(sub2));
259         #fijando los valores mediante el metodo set a total
260         obtenerTotal.set(str(tot1));
261
262     except ValueError:
263         MessageBox.showInfo("AVISO IMPORTANTE","En la columna Cantidad o Precio hay valores incorrectos.
264         Ingrese solo numeros y evite dejar celdas vacias.
265
266         *****
267         | Mensaje de ayuda |
268         *****
269         Puede ingresar solo los codigos de productos y el sistema te arrojará los precios""");
270
271
272
273 def Codigos():
274     """ esta función obtiene por condicionales los valores predeterminado para la casilla codigo de producto, descripcion, unidad y precio"""
275     #Accediendo a los valores mediante el metodo get de codigo, descripcion, unidad y precio
276     try:
277         codg=int(obtenerCodigo.get());
278         codg1=int(obtenerCodigo1.get());
279         codg2=int(obtenerCodigo2.get());
280         desc=obtenerDescripcion.get();
281         desc1=obtenerDescripcion1.get();
282         desc2=obtenerDescripcion2.get();
283
284         prec=obtenerPrecio.get();
285         prec1=obtenerPrecio1.get();
286         prec2=obtenerPrecio2.get();
287
288         uni=obtenerUnidad.get();
289         uni1=obtenerUnidad1.get();
290         uni2=obtenerUnidad2.get();
291
292     #obteniendo valores predeterminados por condicionales para codigo de producto, descripcion, unidad y precio
293
294     #condicion para la primera columna de codigo de producto, precio y unidad
295     if codg>1 and codg<10:
296         desc="Stock"
297         prec=15
298         uni = "Kilogramos"
299     elif codg>10 and codg <30:
300         desc="sin oferta"
301         prec=30
302         uni= "Metros"
303     else:
304         desc="oferta"
305         prec=50
306         uni = "Litros"
307
308     #condicion para la segunda columna de codigo de producto, precio y unidad
309     if codg1>1 and codg1<10:
310         desc1="Stock"
311         prec1=60
312         uni1 = "kilos"
313     elif codg1>10 and codg1 <30:
314         desc1="sin oferta"
315         prec1= 80
316         uni1= "Metros"
317     else:
318

```

Definimos la función calculo, aquí vamos a crear todas las variables de calculo obteniendo valores con get, y asignando valores a otras variables, mediante el calculo de otras variables.

Como el subtotal que es el producto de prec y cant.

Definimos código, aquí vamos a obtener los códigos del id que se ingrese, así como el precio y las unidades de venta.

Después vemos la programación de las cajas de texto que se van a mostrar.

## TRABAJO FINAL DEL CURSO

```

File Edit Selection View Go Run Terminal Help
trabajo final(tornillo)

tornillo_extra.py 9+ X
tornillo_extra.py > Códigos
335 uni2= "Metros"
336
337 #fijando los valores mediante el metodo set a precio, descripcion, codigo y unidad
338 obtenerPrecio.set(str(prec));
339 obtenerPrecio1.set(str(prec1));
340 obtenerPrecio2.set(str(prec2));
341
342 obtenerDescripcion.set(str(desc));
343 obtenerDescripcion1.set(str(desc1));
344 obtenerDescripcion2.set(str(desc2));
345
346 obtenerCodigo.set(str(codg));
347 obtenerCodigo1.set(str(codg1));
348 obtenerCodigo2.set(str(codg2));
349
350 obtenerUnidad.set(str(uni));
351 obtenerUnidad1.set(str(uni1));
352 obtenerUnidad2.set(str(uni2));
353
354 except ValueError:
355     MessageBox.showinfo("AVISO IMPORTANTE","En la columna Cod_Prod ingresar un numero válido.
356 Evite dejas casillas vacias
357
358 NOTA: los códigos de productos solo son números.");
359
360 def Limpiar():
361     """ Esta funcion limpia o reinicia los valores a cero, permitiendo ingresar los valores nuevos por el usuario"""
362     #Accediendo a los valores mediante el metodo get a todos las celdas del segundo frame
363     obtenerCodigo.set("");
364     obtenerCodigo1.set("");
365     obtenerCodigo2.set("");
366     obtenerDescripcion.set("");
367     obtenerDescripcion1.set("");
368     obtenerDescripcion2.set("");
369     obtenerUnidad.set("");
370     obtenerUnidad1.set("");
371     obtenerUnidad2.set("");
372     obtenerCantidad.set("");
373     obtenerCantidad1.set("");
374     obtenerCantidad2.set("");
375     obtenerPrecio.set("");
376     obtenerPrecio1.set("");
377     obtenerPrecio2.set("");
378     obtenerSubtotal.set("");
379     obtenerSubtotal1.set("");
380     obtenerSubtotal2.set("");
381     obtenerTotal.set("");
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Fiamos los valores mediante el método set a precio ,descripción ,código y unidad.

Luego definimos la función limpiar, la cual dejara vacias todas las cajas de texto existentes.

En toda esta parte, era programado el cuerpo de nuestra pantalla principal, los datos para llenar, el estilo de texto y el tamaño, y la ubicación en filas y columnas.

## TRABAJO FINAL DEL CURSO

```

File Edit Selection View Go Run Terminal Help
tornillo_extra.py 9+ X
tornillo_extra.py > ...
434
435 #segundo Frame
436 miFrame1 = Frame(root)
437 miFrame1.pack()
438
439
440
441
442 #nombre de mis variables(CODIGO)
443 obtenerCodigo = StringVar()
444 obtenerCodigo1 = StringVar()
445 obtenerCodigo2 = StringVar()
446 #Label y entry,s Código producto -----
447 lCodigo = Label(miFrame1, text='Cod_Prod')
448 lCodigo.grid(row=9, column=0, sticky='ew', pady=5, padx=5)
449 tCodigo1 = Entry(miFrame1, width=7, textvariable =obtenerCodigo)
450 tCodigo1.grid(row=10, column=0, pady=5, padx=5)
451 tCodigo2 = Entry(miFrame1, width=7, textvariable =obtenerCodigo1 )
452 tCodigo2.grid(row=11, column=0, pady=5, padx=5)
453 tCodigo3 = Entry(miFrame1, width=7, textvariable =obtenerCodigo2)
454 tCodigo3.grid(row=12, column=0, pady=5, padx=5)
455
456
457 #nombres de mis variables (DESCRIPCION)
458 obtenerDescripcion = StringVar()
459 obtenerDescripcion1 =StringVar()
460 obtenerDescripcion2 =StringVar()
461 #Label y entry,s Descripción ----- ESTOY AQUI
462 lDes = Label(miFrame1, text='Descripción')
463 lDes.grid(row=9, column=1, sticky='ew', pady=5, padx=5)
464 tDes1 = Entry(miFrame1, width=7, state="readonly", textvariable = obtenerDescripcion)
465 tDes1.grid(row=10, column=1, pady=5, padx=5)
466 tDes2 = Entry(miFrame1, width=7, state="readonly", textvariable = obtenerDescripcion1)
467 tDes2.grid(row=11, column=1, pady=5, padx=5)
468 tDes3 = Entry(miFrame1, width=7, state="readonly", textvariable = obtenerDescripcion2)
469 tDes3.grid(row=12, column=1, pady=5, padx=5)
470
471 #nombres de mi variable (UNIDAD)
472 obtenerUnidad = StringVar()
473 obtenerUnidad1 = StringVar()
474 obtenerUnidad2 = StringVar()
475 #Label y entry,s Unidad -----
476 lUni = Label(miFrame1, text='Unidad')
477 lUni.grid(row=9, column=2, sticky='ew', pady=5, padx=5)
478 tUni1 = Entry(miFrame1, width=7, state="readonly", textvariable = obtenerUnidad)
479 tUni1.grid(row=10, column=2, pady=5, padx=5)
480 tUni2 = Entry(miFrame1, width=7, state="readonly", textvariable = obtenerUnidad1 )
481 tUni2.grid(row=11, column=2, pady=5, padx=5)
482
483
484
485 #nombres de mi variable (CANTIDAD)
486 obtenerCantidad = StringVar()
487 obtenerCantidad1 = StringVar()
488 obtenerCantidad2 = StringVar()
489 #Label y entry,s Cantidad -----
490 lCantidad = Label(miFrame1, text='Cantidad')
491 lCantidad.grid(row=9, column=3, sticky='ew', pady=5, padx=5)
492 tCantidad1 = Entry(miFrame1, width=7, textvariable = obtenerCantidad)
493 tCantidad1.grid(row=10, column=3, pady=5, padx=5)
494 tCantidad2 = Entry(miFrame1, width=7, textvariable = obtenerCantidad1)
495 tCantidad2.grid(row=11, column=3, pady=5, padx=5)
496 tCantidad3 = Entry(miFrame1, width=7, textvariable = obtenerCantidad2)
497 tCantidad3.grid(row=12, column=3, pady=5, padx=5)
498
499 #nombres de mis variables (PRECIO)
500 obtenerPrecio = StringVar()
501 obtenerPrecio1 = StringVar()
502 obtenerPrecio2 = StringVar()
503 #Label y entry,s Precio -----
504 lPrecio = Label(miFrame1, text='Precio')
505 lPrecio.grid(row=9, column=4, sticky='ew', pady=5, padx=5)
506 tPrecio1 = Entry(miFrame1, width=7, textvariable = obtenerPrecio)
507 tPrecio1.grid(row=10, column=4, pady=5, padx=5)
508 tPrecio2 = Entry(miFrame1, width=7, textvariable = obtenerPrecio1)
509 tPrecio2.grid(row=11, column=4, pady=5, padx=5)
510 tPrecio3 = Entry(miFrame1, width=7, textvariable = obtenerPrecio2)
511 tPrecio3.grid(row=12, column=4, pady=5, padx=5)
512
513 #nombre de mi variable (SUB TOTAL)
514 obtenerSubtotal = StringVar()
515 obtenerSubtotal1 = StringVar()
516 obtenerSubtotal2 = StringVar()
517 #Label y entry,s Subtotal -----
518 lSubtotal = Label(miFrame1, text='Subtotal')
519 lSubtotal.grid(row=9, column=5, sticky='ew', pady=5, padx=5)
520 tSubtotal1 = Entry(miFrame1, width=7, state="readonly", textvariable = obtenerSubtotal)
521 tSubtotal1.grid(row=10, column=5, pady=5, padx=5)
522 tSubtotal2 = Entry(miFrame1, width=7, state="readonly", textvariable = obtenerSubtotal1)
523 tSubtotal2.grid(row=11, column=5, pady=5, padx=5)
524 tSubtotal3 = Entry(miFrame1, width=7, state="readonly", textvariable = obtenerSubtotal2)
525 tSubtotal3.grid(row=12, column=5, pady=5, padx=5)
526
527 #Label y entry,s Total -----
528 obtenerTotal = StringVar() # Nombre de la variable de total
529 lTotal = Label(miFrame1, text='Total', font=("calibri", 12), bg= "#A48D44")
530 lTotal.grid(row=13, column=4, sticky='ew', pady=5, padx=5)
531 tTotal = Entry(miFrame1, width=7, state="readonly", textvariable = obtenerTotal)
532 tTotal.grid(row=14, column=4, sticky='ew', pady=5, padx=5)
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Aquí vemos el tipo de valor que almacena cada variable.

Luego vemos el titulo da las columnas de las cajas de texto, donde programamos la ubicación por filas y columnas, esta programacion va a llevar todas las columnas de caja de texto en la hoja de venta.

## TRABAJO FINAL DEL CURSO

```


File Edit Selection View Go Run Terminal Help
tornillo_extra.py 9+ X
tornillo_extra.py > ...
524 tSubtotal3 = Entry(miFrame1, width=7, state="readonly", textvariable = obtenerSubtotal2)
525 tSubtotal3.grid(row=12, column=5, pady=5, padx=5)
526
527 #Label y entry, s Total -----
528 obtenerTotal = StringVar() # Nombre de la variable de total
529 lTotal = Label(miFrame1, text='Total', font=("calibri", 12), bg= "#A4D8D4")
530 lTotal.grid(row=13, column=4, sticky='ew', pady=5, padx=5)
531 tTotal = Entry(miFrame1, width=7, textvariable = obtenerTotal)
532 tTotal.grid(row=13, column=5, pady=5, padx=5)
533 #Botón guardar -----
534 guardar=Button(miFrame1, text="Calcular", font=("calibri", 15), bg= "#A4D8D4", command=Calculo)
535 guardar.grid(row=15, column=5, pady=10, padx=10)
536
537 #boton Informacion
538 guardar=Button(miFrame1, text='Informacion', font=("calibri", 12), bg= "#A4D8D4", command=Codigos)
539 guardar.grid(row=15, column=3, pady=5, padx=5)
540
541 #boton Limpiar
542 guardar=Button(miFrame1, text='Limpiar', font=("calibri", 12), bg= "red", command= Limpiar)
543 guardar.grid(row=15, column=0, pady=5, padx=5)
544 #menu
545 Mi_Menu = Menu(root)
546 root.config(menu = Mi_Menu)
547
548 submenu = Menu(Mi_Menu, tearoff = 0)
549 Mi_Menu.add_cascade(label = "Archivo", menu = submenu)
550
551 #sub menus
552 submenu.add_cascade(label= "Entrar como Administrador", command = administrador)
553 submenu.add_cascade(label= "Calendario", command = calendario)
554 submenu.add_command(label= "Añadir", command = add)
555 submenu.add_command(label="Información", command = info)
556 submenu.add_separator()
557 submenu.add_command(label="Salir", command = quit)
558
559 #Llamamos a la funcion hora para se ejecute en la ventana raiz de nuestro programa
560 hora()
561 root.mainloop()
562

```

Después de haber terminado con las variables, podemos proceder a crear los botones, que nos permitirá realizar las operaciones de calcular, brindar información y limpiar, todos con la respectiva programación ya antes dicha.

En penultimo lugar, se crea los nombres con los que apareseran las funciones en el menú de la pesatña de arriba.

Y ponemos en ultimo lugar la función hora, para que se ejecute en la raíz y debajo de esta, colocamos `root.mainloop()`

 <b>SENATI</b>	[NOMBRE DEL TRABAJO]	
	[APELLIDOS Y NOMBRES]	[ESCALA]

### LISTA DE RECURSOS

## TRABAJO FINAL DEL CURSO

**INSTRUCCIONES:** completa la lista de recursos necesarios para la ejecución del trabajo.

<b>1. MÁQUINAS Y EQUIPOS</b> pc windows 10.          
<b>3. HERRAMIENTAS E INSTRUMENTOS</b> XAMMP SQLyog Visual Studio Code         
<b>5. MATERIALES E INSUMOS</b> lenguajes: HTML PHP JS SQL       