PROCESO GESTIÓN DEL TALENTO HUMANO

FORMATO INFORME MENSUAL EJECUCIÓN CONTRACTUAL

Montería, 10-2023

Señor (a)
Nasli Lopez Castellar
SUPERVISOR(A) CONTRATO No. CO1.PCCNTR.5204213
Coordinadora Académica
Centro de Comercio, Industria y Turismo de Córdoba
Montería

Asunto: Informe mensual de ejecución contractual
Mes octubre del año 2023

Referencia: No. CO1.PCCNTR.5204213 del año 2023.

ALFONSO JOSE PATERNINA DIAZ, identificado con la cédula de ciudadanía No. 78.715.466 de Montería, en mi calidad de Contratista del SENA, en el Centro de Comercio, Industria y Turismo de Córdoba, en cumplimiento del Contrato de Prestación de Servicios de la referencia, a continuación, presento el Informe de actividades realizadas en el mes objeto de cobro.

Valor y forma de Pago: Se fija como valor total para el contrato la suma de CATORCE MILLONES OCHOCIENTOS MIL PESOS M/CTE. ($14.800.000). Esta suma será pagada por el SENA al contratista de la siguiente manera: a) CUATRO (4) pagos iguales por los meses de agosto a noviembre de 2023, por valor de TRES MILLONES SETECIENTOS MIL PESOS M/CTE. ($3.700.000) cada uno.

Plazo: Será hasta el 30 de noviembre de 2023.

OBJETO: CONTRATAR LOS SERVICIOS PERSONALES DE CARÁCTER TEMPORAL COMO INSTRUCTOR CONTRATISTA PARA IMPARTIR FORMACIÓN PROFESIONAL INTEGRAL EN EL MARCO DEL PROGRAMA OFERTA REGULAR, TITULADA Y COMPLEMENTARIA PRESENCIAL Y VIRTUAL EN EL CENTRO DE COMERCIO INDUSTRIA Y TURISMO DEL SENA REGIONAL CÓRDOBA

Obligaciones Especificas:

No	Obligaciones	Acciones realizadas	Evidencias
1	Apoyar la Planeación y ejecución la formación profesional de acuerdo con el programa de formación asignado, para el logro de los resultados de aprendizaje definidos cumplido las horas de formación de las competencias de los programas de formación equivalentes a 160 horas mensuales.	Se ejecuta formación profesional integral en las fichas tituladas: 2617684,2503221,2712269	Lista de asistencia, formato mensual de eventos, fotografías, diapositivas, actividades,
2	Colaborar con la elaboración de la planeación pedagógica, guías de aprendizaje, instrumentos de evaluación, plan de trabajo, ayudas didácticas requeridos para el desarrollo de los procesos de formación asignados.	Se elaboraron guías de aprendizaje para las fichas 2617684,2503221,2712269	Guías de aprendizaje, diapositivas, lista de asistencia
3	Registrar en el aplicativo SOFIA PLUS, todas las evidencias del proceso formativo: crear las rutas de aprendizaje, asociar aprendices, realizar registro de inasistencia de aprendices, planes de mejoramiento y registrar juicios evaluativos, acorde con los lineamientos institucionales.	Una vez se asignen fichas en el Aplicativo Sofia Plus, se realizarán los respectivos juicios evaluativos	N/A
4	Presentar las actas finales de cierre de etapa lectiva de las fichas asignadas como responsable, acorde con los lineamientos institucionales	A la fecha no se asignados fichas que requieran cierre de etapa lectiva	N/A
5	Apoyar el trámite de las novedades de deserción de aprendices acorde con lo	No se han presentados deserciones	N/A

	establecido en el reglamento del aprendiz		
6	Asistir a los aprendices en las novedades acorde con lo establecido en el reglamento del aprendiz	Se orientará y asistirá al aprendiz una vez se requiera	N/A
7	Aportar todos los documentos soporte del proceso de formación profesional, publicados en la plataforma SIGA.	Guías de aprendizaje desarrolladas en el mes para las fichas 2617684,2503221,2712269	Guías de aprendizaje
8	Participar en proyectos de investigación aplicada, técnica y pedagógica en función de la formación profesional de los programas relacionados con el área temática	A la fecha no se han asignado actividades de este tipo	N/A
9	Subir los documentos soporte del pago en la plataforma del SECOP II en los plazos establecidos por la entidad y acorde con los requisitos legales exigidos.	Se realizó el cargue de los documentos en el aplicativo SECOP2 de las carpetas GF y GC y el informe de ejecución mensual	Formato informe mensual de ejecución GTH-F-062 V10
10	Comunicar al Supervisor del contrato en un plazo máximo de tres (3) días hábiles anomalías, inconsistencias, novedades de aprendices y hallazgos en el registro de la información.	Se realizarán las respectivas notificaciones cada que se requiera	N/A
11	Comunicar al Supervisor del contrato en un plazo no inferior a quince (15) días hábiles las anomalías, inconsistencias, novedades del Contrato (cesión - Terminación).	Se realizarán las respectivas notificaciones cada que se requiera	N/A
12	Cumplir con los lineamientos del sistema de gestión de SST y los lineamientos establecidos por el gobierno nacional frente la pandemia del COVID 19.	Se realizó curso introductorio de SST Se atienden los lineamientos del grupo de salud y seguridad en el trabajo	Certificado curso introductorio Se mantiene el distanciamiento físico, lavado de

			manos y uso de la mascarilla
13	Atender las solicitudes realizadas por el supervisor del contrato.	Se remiten los eventos mensuales, horarios individuales del instructor,	Formato de eventos Horario del instructor
14	Gestionar la colocación de aprendices en alguna de las modalidades de etapa productiva y rendir informes periódicos de los resultados alcanzados, conforme a las instrucciones que les sean impartidas.	No se ha requerido realizar actividades de este tipo	N/A

A continuación, relaciono los desplazamientos que realicé previo a la presentación de este informe. Una vez finalizado cada desplazamiento presenté al ordenador del gasto el informe en el Formato Informe Legalización Desplazamiento Contratista GTH-F-087, en el que se describieron las actividades desarrolladas y los resultados de cada desplazamiento. Cada informe cuenta con el visto bueno del Supervisor.

Se lista a continuación el soporte de la legalización de los desplazamientos realizados, los cuales forman parte integral del presente informe de ejecución contractual.

ITEM	No DE LA ORDEN DE VIAJE	LUGAR DE DESPLAZAMIENTO	FECHA DE DESPLAZAMIENTO INICIAL	FECHA DE DESPLAZAMIENTO FINAL
1.	XX	XX	XX	XX
2.	XX	XX	XX	XX

Nota 1: Por cada desplazamiento que haya realizado el contratista, adjuntará el respectivo informe que la soporte. En caso de haber realizado el desplazamiento en fecha posterior a la presentación del informe de ejecución contractual, deberá reportarlo en el siguiente informe de ejecución contractual.

Para el trámite de la cuenta me permito adjuntar: Documentos electrónicos enunciados como evidencias del cumplimiento de las obligaciones contractuales y los desplazamientos realizados y el No.4518283972 de la planilla, operador SOI y periodo septiembre de 2023. (Decreto Ley 2106 de 2019 – “Decreto Ley Antitrámites”)

Evidencias en (42) folios

Cordialmente,

Alfonso José Paternina Díaz
Contratista
C.C. No. CO1.PCCNTR.5204213.

Recibí a satisfacción:

Nasly Lopez Castellar
Supervisor(a) Contrato Nº CO1.PCCNTR. 5204213 de 2023
Coordinadora Académica

REPORTE MENSUAL DEL INSTRUCTOR

REGIONAL CORDOBA
CENTRO DE COMERCIO, INDUSTRIA Y TURISMO

DATOS DEL INSTRUCTOR		MES:	OCTUBRE	AÑO:	2023
NOMBRES Y APELLIDOS:	ALFONSO JOSE PATERNINA DIAZ	CÉDULA:	78715466		

PARTE HORAS DIRECTAS A FORMACIÓN - PROGRAMAS EN FORMACIÓN TITULADA Y/O COMPLEMENTARIA

	PROGRAMA FORMACIÓN					HORARIO		FECHA		DIAS							COMPETENCIA DESARROLLADA	RESULTADOS DE APRENDIZAJE	HORAS MES	FINALIZACION				
	No. FICHA	PROGRAMA	NIVEL DE FORMACION	MUNICIPIO	AMBIENTE	DESDE	HASTA	DESDE	HASTA	L	M	M	J	V	S	D				HORAS TOTAL DE LA COMPETENCIA	HORA MES ANTERIOR	HORAS PENDIENTE	FINALIZA
	2712269	PROGRAMACION DE SOFTWARE	TECNICO	SAN BERNARDO DEL VIENTO		12:00	20:00	3-oct	31-oct		X	X	X				DESARROLLAR LA SOLUCIÓN DE SOFTWARE DE ACUERDO CON EL DISEÑO Y METODOLOGÍAS DE DESARROLLO	CODIFICAR EL SOFTWARE EMPLEANDO EL LENGUAJE DE PROGRAMACION SELECCIONADO	104					
4	2503221	ANALISIS Y DESARROLLO DE SOFTWARE	TECNICO	MONTERIA	M - 105	16:00	22:00	6-oct	27-oct					X			Evaluar requisitos de la solucion de softwarede acuerdo con metologias de analisis y estandares	MODELAR LAS FUNCIONES DEL SOFTWARE DEACUERDO CON EL INFORME DE REQUISITOS	24					
7	2617684	ANALISIS Y DESARROLLO DE SOFTWARE	TECNICO	MONTERIA	C - 201	12:00	20:00	7-oct	28-oct						X		DESARROLLAR LA SOLUCIÓN DE SOFTWARE DE ACUERDO CON EL DISEÑO Y METODOLOGÍAS DE DESARROLLO	CONSTRUIR LA BASE DE DATOS PARA EL SOFTWARE A PARTIR DEL MODELO DE DATOS	32					
																		TOTAL HORAS	160					

PARTE HORAS OTRAS ACTIVIDADES

PARTE APOYO PRESTADO ACTIVIDADES ADMINISTRATIVAS

	ACTIVIDAD	JUSTIFICACIÓN	FECHA DESDE	FECHA HASTA	HORARIO DESDE	HORARIO HASTA	HORAS
1							
2							
3							
4							
5							
				TOTAL HORAS			0

PARTE CAPACITACIÓN INSTRUCTORES PLANTA

	NOMBRE DE LA CAPACITACIÓN	FECHA DESDE	FECHA HASTA	HORARIO DESDE	HORARIO HASTA	HORAS
1						
2						
3						
			TOTAL HORAS			0

PARTE OTRAS ACTIVIDADES INSTRUCTORES PLANTA

			FECHA DESDE	FECHA HASTA	HORARIO DESDE	HORARIO HASTA	HORAS
1	Preparación de clases						
2	Incapacidad						
3	Lider WorldSkills						
4	Permiso Sindical						
5	Investigacion						
6	Permiso por estudio						
7	Act. Culturales (Evento deportivo Nacional)						
					TOTAL HORAS		0

TOTAL HORAS REPORTADAS 160

FIRMA DEL INSTRUCTOR:

EVIDENCIAS

Lista de asistencia Programacion de software ficha 2712269

Sistema de Gestión de la Calidad

Servicio Nacional de Aprendizaje SENA
Centro de Comercio, Industria y Turismo de Córdoba
CONTROL DE ASISTENCIA DIARIA
FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.

Nombre del Curso:	Programacion de Software		
FICHA No.	2712269		
Municipio:	San Bernardo del Viento	Lugar:	
Instructor:	Alfonso Jose Paternina Diaz	Nº. Doc. Ident.	
FECHA:	12-10-2023	HORARIO:	12:00 - 20:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Chiquillan	Jesi Camilo	1068873972	
02	Diaz Diaz	Alexander	1070807822	
03	Guevara Diaz	Andres Felipe	1003562103	Andres
04	Guerrero de los R.	Maria Liz	1070811780	Maria.
05	Batista Padilla	Julian Jose	1067955286	
06	Quintana Pajaro	Johanis de C.	50970615	
07	Alvarez Suarez	Juan David	1068736952	Juan.D
08	Cardenas Lopez	Kevin Jose	1070810888	Kevin C.
09	Isaza Mejia	Maria Jose	1070809508	Maria Jose.
10	Velasquez perez	Naira Marcela	1068737851	Marcela
11	Perez Garcia	Luis Felipe	1070809627	Luis Felipe
12	Martinez Ballesta	Lucia de Jesus	26138197	Lucia M.B.
13	Fuentes Castro	Samer Jose	1070815603	Samer.F.C
14	Fuentes Arrieta	Jerson	1063734621	Jerson
15	Barcasnegras Viafara	Carlos Javier	1112175627	Carlos Javier B V.
16	Mejia Martelo	Daniel Enrique	1003562340	Daniel Mejia
17	Pepo Burgos	Donaldo	1003779230	
18	Pedro Burgos	Jose David	1063784946	
19	Carmona de Hoyos	Rosa del carmen	1070818238	
20	Ballesta Vidal	Jaime de Jesus	1068736900	
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Paternina Diaz

Sistema de Gestión de la Calidad

Servicio Nacional de Aprendizaje SENA
Centro de Comercio, Industria y Turismo de Córdoba
CONTROL DE ASISTENCIA DIARIA
FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.

Nombre del Curso:	Programacion de Software		
FICHA No.	2712269		
Municipio:	San Bernardo del Viento	Lugar:	
Instructor:	Alfonso Jose Paternina Diaz	Nº. Doc. Ident.	
FECHA:	04-10-2023	HORARIO:	

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Guerrero De Jose	Maria J13	1070811780	Maria Guer.
02	Alvarado Diaz	Andres felipe	1003562103	Andres Alvarado
03	Quintana Pajaro	Johanis del C.	50970615	
04	Batista Padilla	Julian Jose	1067952867	
05	Perez Garcia	Luis Felipe	1070809227	Luis Felipe
06	Martinez Ballesta	Lucia de Jesus	26138197	Lucia martinez
07	Velasquez Perez	Naira Marcela	1068737851	Marcela
08	Orango Romero	Brigith Margarita	1068738408	Brigith M.
09	Isaza Mejia	Maria Jose	1070809508	Maria Jose
10	Cardenas Lopez	Kevin Jose	1070810388	Kevin C.
11	Clica Cera	Jose Camilo	1070	Camilo
12	Diaz Diaz	Alexander	7070807870	
13	Barcenegra Victoria	Carlos Javier	1192705627	Carlos Javier B
14	Fuentes Castro	Samir Jose	1070825603	Samir
15	Pacheco Cardenas	Maria Isabel	1070810599	Maria Pacheco
16	Franco Paredes	Leiner Luis	1070809628	Leiner franco
17	Negrete Herrera	Greisy Paola	1066936729	Greisy
18	Carmona de hoyos	Rosa del Carmen	1070813239	Rosa Carmona
19	Ballesta Vitar	Jaime de Jesus	1066736700	Jaime Ballesta
20	Fuentes Arrieta	Jerson	1063729609	Jerson
21	Media Morelo	Daniel Enrique	1003562340	Daniel Media
22	Aguilar Suarez	Juan David	1068736952	Juan D
23	Petro Burgos	Donaldo	1063749230	Donaldo
24	Petro Borgos	Jose Daniel	1063748	
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Paternina Diaz

Sistema de Gestión de la Calidad	**Servicio Nacional de Aprendizaje SENA** Centro de Comercio, Industria y Turismo de Córdoba **CONTROL DE ASISTENCIA DIARIA** **FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.**

Nombre del Curso:	Programacion de Software		
FICHA No.	2712269		
Municipio:	San Bernardo del Viento	Lugar:	
Instructor:	Alfonso José Paternina Diaz	Nº. Doc. Ident.	
FECHA:	11-10-2023	HORARIO:	12:00 - 20:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Quintana Pajaro	Johanis	50970615	
02	Batista Padilla	Julian Jose	1062952867	
03	Guevara Diaz	Andres Felipe	1003562103	Andres Guevara
04	Velasquez Perez	Naira Marcela	1068737851	Marcela
05	Isaza Mejia	Maria Jose	1070809508	Maria Jose
06	Cardenas lopez	Kevin Jose	1070810388	Kevin C.
07	Algamez Suarez	Juan David	1068736952	
08	Diaz Diaz	Alexander	1070807870	
09	Fuentes Castro	Samir Jose	1070815603	Samir F. C
10	Pacheco Cardenas	maria isabel	1070810599	Maria Pacheco
11	Franco Paredes	Leiner luis	1070809588	Leiner Franco
12	Fuentes Arrieta	Jerson	1063739629	Jerson
13	Carmona de hoyos	rosa del Carmen	1090813239	Rosa
14	Bellesteu Vitola	Jaime de Jesus	1066736700	Jaime Bellesteu.
15	Mejia Morelo	Daniel Enrique	1003562340	Daniel Mejia
16	Petro Burgos	Donaldo	1063774230	
17	Petro Burgos	Jose Daniel	1063789446	
18	Perez Enrique	Luis Felipe	1070809227	Luis Felipe
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso José Paternina Diaz

Sistema de Gestión de la Calidad

Servicio Nacional de Aprendizaje SENA
Centro de Comercio, Industria y Turismo de Córdoba
CONTROL DE ASISTENCIA DIARIA
FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.

Nombre del Curso:	Programacion de Software		
FICHA No.	2712269		
Municipio:	San Bernardo del Viento	Lugar:	
Instructor:	Alfonso Jose Paternina Diaz	Nº. Doc. Ident.	
FECHA:	10-10-2023	HORARIO:	12:00 - 20:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Barrasuegos Viafara	Carlos Javier	1112915627	Carlos Javier B V
02	Fuentes Arrieta	Jerson	1063734624	Jerson
03	Ballesta Vitar	Jaime de Jesus	1068736700	Jaime Ballesta.
04	Carmona de Lopez	Rosa del Carmen	1070813234	Rosa Carmona
05	Franco Paredes	Leiner Luis	1070809548	Leiner Franco
06	Pacheco Cardenas	maria Isabel	1070810599	Maria Pacheco
07	Fuentes Castro	Samir jose	1070815603	Samir
08	Diaz Diaz	Alexander	1070807870	Alexander
09	Chica Beltran	Jose Camilo	1070809972	Camilo
10	Pedro Burgos	Donaldo	1063718496	Donaldo.
11	Pedro Burgos	José Daniel	1063718496	
12	Higamez Suarez	Juan David	1068736952	
13	Quintana Pajaro	Johanis del C	50970615	
14	Batista Padilla	Julian Jose	1067752841	
15	Guerrero Diaz Reyes	Maria Jose	1070811780	Maria
16	Guevara Diaz	Andres Felipe.	1003562103	Andres
17	Negrete Herrera	Greisy Paola	1065730729	Greisy N
18	Ibaza Mejia	Mayra Jose	1070809508	Mayra Jose
19	Cardenas Lopez	Kevin jose	1070810388	Kevin C
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Paternina Diaz

Sistema de Gestión de la Calidad	**Servicio Nacional de Aprendizaje SENA** Centro de Comercio, Industria y Turismo de Córdoba **CONTROL DE ASISTENCIA DIARIA** **FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.**

Nombre del Curso:	Programacion de Software		
FICHA No.	2712269		
Municipio:	(Monteria) Sanbernardo del Viento	Lugar:	
Instructor:	Alfonso Jose Paternina Diaz	Nº. Doc. Ident.	
FECHA:	28-09-2023	HORARIO:	12:00 - 20:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Barcasnegras Viafara	Carlos Javier	1112475627	Carlos Javier BV.
02	Carmona de Hoyos	rosa del Carmen	1070813234	Rosa Carmona
03	Ballesteros Vitar	Jaime de Jesus	1067356711	Jaime Ballesteros
04	Fuentes Arrieta	Jerson	1063739629	Jerson
05	Franco Paredes	Leiner Luis	1070809520	Leiner Franco
06	Pacheco cardenas	maria isabel	1010910599	Maria Pacheco
07	fuentes castro	Samir josé	1070815603	Samir F
08	Diaz Diaz	Alexander	1070807870	
09	Chica Cera	José camilo	1070808772	
10	Pedro Burgos	Donaldo	1063719230	
11	Pedro Burgos	José Danicl	1065718446	
12	Agamez Suarez	Juan pard	1068736952	Juan D.
13	Quintana Pajaro	Johanis del C	50 970 615	
14	Batista Padilla	Julian Jose	1067958283	
15	Guerra Reyes	Maria Fe	1010811780	Maria
16	Guerra Diaz	Andres Felipe	1003502103	Andres
17	Negrete Herrera	Greisy Paola	1065736779	Greisy N.
18	Isaza Mejia	Maria Jose	1070809588	Maria Jose
19	Cardenas Lopez	Kevin Jose	1010810388	Kevin C.
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Paternina Diaz

Sistema de Gestión de la Calidad	**Servicio Nacional de Aprendizaje SENA** Centro de Comercio, Industria y Turismo de Córdoba **CONTROL DE ASISTENCIA DIARIA** **FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.**

Nombre del Curso:	Programacion de Software		
FICHA No.	2912269		
Municipio:	San Bernardo del Viento	Lugar:	
Instructor:	Alfonso Jose Patrinina Diaz	Nº. Doc. Ident.	
FECHA:	26-09-2023	HORARIO:	12:00 - 20:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Guevara Diaz	Andres Felipe	1003562103	Andres
02	Guerrero D.t. Reyes	Maria Lis	1070811760	Maria
03	Negrete Herrera	Gregori Pedro	1068738729	
04	Isaza Mejia	Maria Jose	1070889508	Maria Jose
05	Quintana Pajaro	Johanis	50970613	
06	Batista Peldilla	Julian Jose	1067952861	
07	Cardenas Lopez	Kevin Jose	1070810388	Kevin C.
08	Agamez Suarez	Juan Daniel	1068736152	Juan D
09	Cortes Gomez	Jose Camilo	1070809792	Camilo
10	Cales Javier	Barcasnegras Viafara	1112475627	Carlos Javier BV
11	Fuentes Castro	Samer Jose	1070815603	Samer F.C
12	Franco Paredes	Leiner Luis	1070809458	Leinfranco
13	Bolleda Vitola	Jaime de Jesus	1068736100	Jaime B.
14	Fuentes Arrieta	Jerson	1063739621	Jerson
15	Media Morelo	Daniel Enrique	1003562340	Daniel M.
16	Petro Burgos	Donaldo	1063719730	Donaldo
17	Petro Burgos	Jose Daniel	1068718446	
18	Perez Garcia	Luis Felipe	1070809227	Luis Felipe
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Patrinina Diaz

Sistema de Gestión de la Calidad

Servicio Nacional de Aprendizaje SENA
Centro de Comercio, Industria y Turismo de Córdoba
CONTROL DE ASISTENCIA DIARIA
FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.

Nombre del Curso:	Programacion de Software		
FICHA No.	2712269		
Municipio:	San Bernardo del Viento	Lugar:	
Instructor:	Alfonso Jose Patrinina Diaz	Nº. Doc. Ident.	
FECHA:	27-09-2023	HORARIO:	12:00 - 20:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Agamez Suarez	Juan Pavol	1068736952	JuanD
02	Cardenas lopez	Kevin jose	1070810388	Kevin C.
03	Bautista Padilla	Julian Jose	1067728284	
04				
05	Isaza Tejeda	Maria Jose	1070809508	Maria Jose
06	Urango Romero	Brigith Margarita	1068738408	Brigith M
07	Velasquez Perez	Naira Marcela	1068737851	Marcela
08	Negrete Herrera	Evelyn Paola	1068736729	
09	Perez Garcia	Luis Felipe	1070809627	Luis Felipe
10	Martinez Ballesta	Lucia de Jesus	26138197	Lucia M B
11	Guevara Diaz	Andres Felipe	1003562767	Andres
12	Guerrero de la R.	Maria J.	1070811780	
13	Media Morelo	Daniel Enrique	1003562340	Daniel Media
14	Barcasnegras Viafara	Carlos Javier B	1112475627	Carlos Javier B V.
15	Fuentes Arrieta	Jerson	1063719629	Jerson
16	Ballesta Vitar	Jaime de Sesus	1066736700	Jaime Ballesta.
17	Carmona de hoyos	Rosa del Carmen	1070813224	Rosa Carmona
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Patrinina Diaz

www.sena.edu.co
SENAComunica

Sistema de Gestión de la Calidad

Servicio Nacional de Aprendizaje SENA
Centro de Comercio, Industria y Turismo de Córdoba
CONTROL DE ASISTENCIA DIARIA
FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.

Nombre del Curso:	Programacion de Software		
FICHA No.	2712269		
Municipio:	San Bernardo del Viento	Lugar:	
Instructor:	Alfonso Jose Paternina Diaz	Nº. Doc. Ident.	
FECHA:	05/10/2023	HORARIO:	12:00 - 20:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Quintana Pajaro	Johanis del C.	50970615	
02	Batista Padilla	Julian Jose	1067952863	
03	Guevara Diaz	Andres Felipe	1003562103	
04	Chica Cenes	Jose Camilo	1078009772	
05	Velasquez Perez	Naiga Marcela	1067737851	
06	Cardenas Lopez	Kevin Jose	1070810388	
07	Isaza Mejia	Manuel Jose	1070809508	
08	Agamez Suarez	Juan David	1068736952	
09	Martinez Ballesta	Lucia de Jesus	26138197	
10	~~Perez Garcia~~	Luis Felipe	1070509227	
11	Fuentes Arrieta	Jerson	1063734629	
12	Guerrero de Reyes	Maria Luis	1070811780	
13	Carmona de Hoyos	Rosa del Carmen	1070813234	
14	Ballesta Vitar	Jaime de Jesus	1068736200	
15	Barcasnegras Viafara	Carlos Javier	1142945627	
16	Mejia Morelo	Daniel Enrique	1003562340	
17	Pitre Barros	Donaldo	1063719730	
18	Negrete Herrera	Greisy Paola	1068736729	
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Paternina Diaz

Sistema de Gestión de la Calidad	**Servicio Nacional de Aprendizaje SENA** Centro de Comercio, Industria y Turismo de Córdoba **CONTROL DE ASISTENCIA DIARIA** **FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.**

Nombre del Curso:	Programacion de Software		
FICHA No.	2712769		
Muaicipio:	San Bernardo del Viento	Lugar:	
Instructor:	Alfonso Jose Paternina Diaz	N°. Doc. Ident.	
FECHA:	03 - 10 - 2023	HORARIO:	12:00 - 20:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Guerrero D. L. Reyes	Maria Liz	1010811780	Maria G
02	Guevara Diaz	Andres Felipe	1003562168	Andres.
03	Batista Padilla	Julian Jose	1067952882	
04	Quintana Papin	Johanis del C	50970615	
05	Cardenas lopez	Kevin Jose	1010860800	Kevin C.
06	Espeja Mejia	Maria Jose	1070809508	Maria Jose
07	Negrete Herrera	Greisy paola	1068736929	Greisy N
08	Martinez Ballesta	Lucia de Jesus	26138197	Lucia M B.
09	Urango Romero	Brigith Margarita	1068738408	Brigith M
10	Perez Garcia	Luis Felipe	1070809727	Luis Felipe
11	Velasquez Perez	Naira Marcela	1068737851	Marcela.
12	Franco Paredes	Leiner Luis	1070804588	LeinerFranco
13	Pacheco Cardenas	Maria Isabel	1010810599	Maria Pacheco
14	Samir José	Fuentes Castro	1070816073	Samir. Fc
15	Chica Coy	Jose Carril	1070808772	Carril
16	Diaz Diaz	Alexander	1070807870	
17	Jaramillo Suarez	Juan David	1018736952	Juan. D
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Paternina Diaz

EVIDENCIAS

Lista de asistencia Analisis y desarrollo de software ficha 2503221

Sistema de Gestión de la Calidad

Servicio Nacional de Aprendizaje SENA
Centro de Comercio, Industria y Turismo de Córdoba
CONTROL DE ASISTENCIA DIARIA
FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.

Nombre del Curso:	Python Analisis y Desarrollo de Software		
FICHA No.	2503221		
Municipio:	Montería	Lugar:	CCIT
Instructor:	Alfonso Patarroyo	Nº. Doc. Ident.	
FECHA:	15/09/2023	HORARIO:	16:00 - 22:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Cordero Bustamante	Harold Daniel	1067930741	
02	Figueroa Gonzalez	Sergio David	1067950103	
03	Ubarne Doria	Kathy Marcela	1067947191	
04	Massey Perez	Daniela Alejandra	1002498397	
05	Mejia Vega	Javier David	1003432976	
06	Hansares Florez	Maria Camila	1029720403	
07	Pereira Ortiz	Silvia Estela	1029034072	
08	Posso Ospino	Julio Enrique	1062429904	
09	Bueños Bueños	Roberto Miguel	1071354424	
10	Ubarnes Uranzo	Carlos	1068580875	
11	Salgado Otero	Angel David	1062432696	
12	Espinoza Orozco	Sergio M.	80960343	
13	Lopez Falon	Juan Manuel	1064978605	
14	Polo Quiñones	Argemiro Polo	1067954419	
15	Padilla Urango	Sebastian	1067842499	
16	Mogollon Mendoza	Mishell	1063807741	
17	Padilla Tavalera	Roberto Carlos	1067884063	
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Patarroyo Diaz

Sistema de Gestión de la Calidad

Servicio Nacional de Aprendizaje SENA
Centro de Comercio, Industria y Turismo de Córdoba
CONTROL DE ASISTENCIA DIARIA
FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.

Nombre del Curso:	Analisis y desarrollo de Software		
FICHA No.	2503221		
Municipio:	Montería	Lugar:	
Instructor:	Alfonso Jose Petronina D.	Nº. Doc. Ident.	
FECHA:	13 - 10 - 2023	HORARIO:	16:00 - 22:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Polo Quiñones	Argemiro Jose	1067959419	Argemiro Polo
02	Pertuz Jimenez	Roberto Miguel	1071384421	
03	Lopez Falon	Juan Manuel	1064978605	Juan Lopez
04	Posso Osorno	Julio Enrique	1062424909	
05	Mosseri Perez	Daniela Alejandra	1002748392	Daniela M.
06	Padilla Urango	Sebastian	1067842494	Sebastian P.
07	Perez Banos	Manuel Jose	1003434944	
08	Ubarnes Urango	Carlos	1068580875	Carlos U.
09	Espinosa Pulgo	Jorge Mario	8960343	Jorge Omero
10	Dias Rodriguez	Juan Jose	1067838494	Juan J.
11	Delgado Brito	Angel David	1062432694	
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Petronina Diaz

Sistema de Gestión de la Calidad

Servicio Nacional de Aprendizaje SENA
Centro de Comercio, Industria y Turismo de Córdoba
CONTROL DE ASISTENCIA DIARIA
FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.

Nombre del Curso:	Analisis y Desarrollo de Software		
FICHA No.	2503221		
Municipio:	Montería	Lugar:	
Instructor:	Alfonso Jose Patrinina Diaz	Nº. Doc. Ident.	
FECHA:	6-10 - 2023	HORARIO:	16:00 - 22:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Figueroa Gonzalez	Sergio David	1.067.930103	Sergio Figueroa
02	Luis Alfonso Negrete Chamy	Luis Alfonso	1067905794	
03	Magallon Mendoza	Michell	1068807741	Michell M.
04	Padilla Urango	sebastian	1067842494	sebastian padilla
05	Pacilla T	Rogerio Carlos	1062884063	RCPT
06	Bulgado Otero	Angel David	1062452696	Angel.
07	Ubarne Dona	Kathy Marcela	1067947161	Kathy Ubarne
08	Cordero Betancourt	Harold Daniel	1067930741	
09	Mejia Vega	daider David	1003432976	
10	Lopez Falon	Juan Manuel	1064978605	Juan Lopez
11	Polo Quinones	Argemiro Jose	1067954419	Argemiro Polo.
12	Espinoso Orozco	Jorge M	80760343	Jorge Orozco
13	Ubarnes Urango	Carlos	1068580875	
14	Pinto Cortina	Julio Enrique	1062429907	
15	Diaz Rodriguez	Juan Jose	1067838494	Juan Diaz
16	Mussely Perez	Daniela Alejandra	1002798591	Daniela M
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Patrinina Diaz

EVIDENCIAS

Lista de asistencia Analisis y desarrollo de software ficha 2617684

Sistema de Gestión de la Calidad

Servicio Nacional de Aprendizaje SENA
Centro de Comercio, Industria y Turismo de Córdoba
CONTROL DE ASISTENCIA DIARIA
FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.

Nombre del Curso:	Analisis Desarrollo de Software		
FICHA No.	2617684		
Municipio:	Montería	Lugar:	
Instructor:	Alfonso Jose Patrimina Diaz	Nº. Doc. Ident.	
FECHA:	07-10-2023	HORARIO:	12:00 - 20:00

Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Hernández Lora	Carlos Andres	10784113	Carlos Hernández
02	Alvarez Tapia	July Karina	1003398248	July Alvarez
03	Durango Gonzalez	Oscar Dario	1091349121	
04	Saont Altamiranda	Jesus David	1062431510	
05	Borja Nerio	Jenson Javier	1062939347	
06	Padilla Diaz	Kevin Javier	1062951114	Kevin Padilla
07	Salazar Almario	Robinson Mateo	1062965033	Robinson Salazar
08	Pestana Diaz	Jaiber David	1003305964	
09	Hernandez Montes	Santiago	1016006882	Santiago H.
10	Coronado Martinez	Jesús David	1062959097	Jesús Coronado
11	Cavadia Correa	Paula Andrea	1127587786	
12	Patrimina Altamiranda	Mildreis Andrea	1003003576	Mildreis P.
13	Narvaez Sitt	Thandra	1003407303	
14	Alarcon Nieto	Harold Javier	1062435010	
15	Cardozo Camacho	Yeickon Stiven	1094927866	Yeickon
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Patrimina Diaz

Sistema de Gestión de la Calidad

Servicio Nacional de Aprendizaje SENA
Centro de Comercio, Industria y Turismo de Córdoba
CONTROL DE ASISTENCIA DIARIA
FORMATO ÚNICO PARA REPORTE DE ASISTENCIA CCIT.

Nombre del Curso:	Analisis Desarrollo de Software		
FICHA No.	2617684		
Municipio:	Montería	Lugar:	
Instructor:	Alfonso Jose Paternina Diaz	Nº. Doc. Ident.	
FECHA:	14-10-2023	HORARIO:	12:00 - 20:00

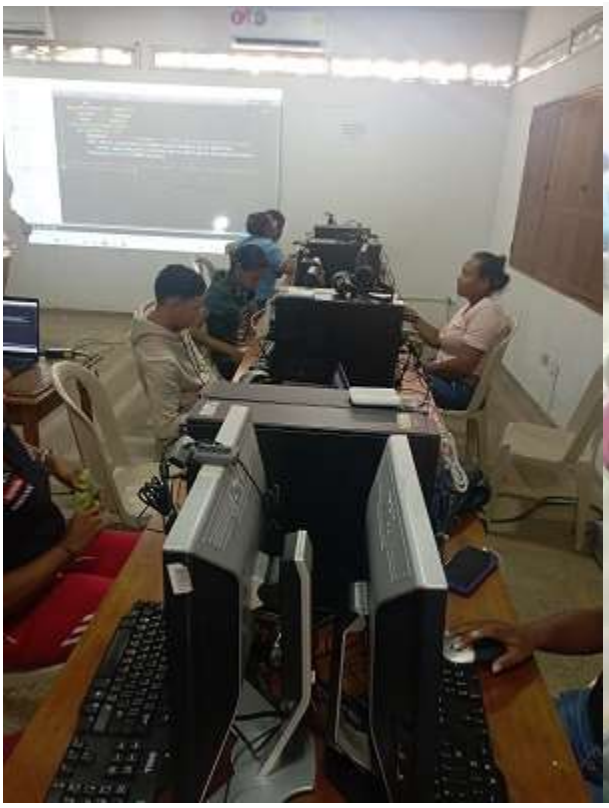
Nº	APELLIDOS	NOMBRES	DOC. IDENT.	FIRMAS
01	Alvarez Tapia	July Karina	1003398248	July Alvarez
02	Paternina Altamiranda	Mildres Andrea	1003003576	Mildres P.
03	Suarez Altamiranda	Jesus David	1061431510	Jesus
04	Morillo Rodriguez	Luis Angel	1067955573	
05	Olivero Pacheco	Mateo	1067787983	Mateo olivero
06	cuadrado carvajal	Nayis isabel	1064317439	Nayis Isabel
07	Hernandez Lara	Carlos Andres	1078411?	Carlos Hernandez
08	Rojo Nevio	Jeison Javier	1063754350	
09	Cogollo Vertel	David	1067.946.903	
10	Salazar Almario	Robinson Miguel	1062965033	Robinson S.
11	Narvaez Soto	Jhondzz	1003407303	JN
12	Alarcon Nieto	Harold Duvan	1062433010	
13	Rava Seria	Angel David	1003431882	Angel Rava
14	Cardozo Camacho	Yeickon Stick	1099427888	Yeickon S.
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				

FIRMA DEL INSTRUCTOR Alfonso Jose Paternina Diaz

Evidencias Fotograficas de Programacion de Software Ficha(2712269)









Evidencias Fotograficas Analisis y desarrollo de software ficha (2503221)









Evidencias Fotograficas de Analisis y desarrollo de software ficha(2617684)







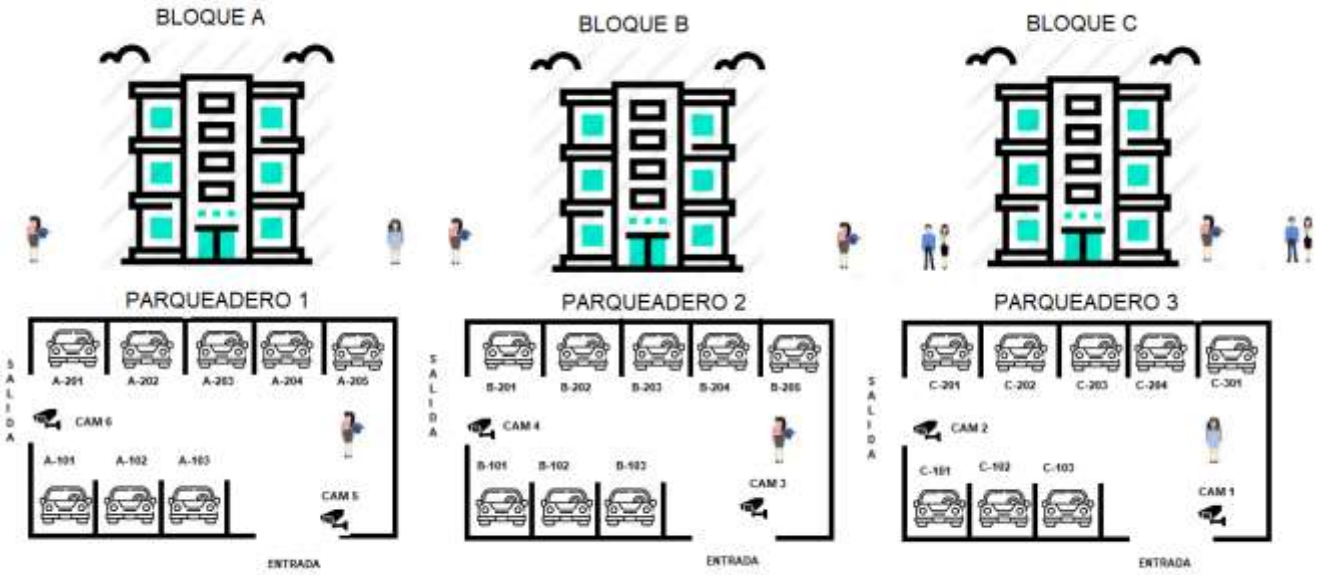


REQUERIMIENTOS DEL PROTOTIPO

Conjunto residencial Brisas del mar está ubicado en la cra 45 No 124-12 del Municipio de Puerto Escondido, cuenta con 3 bloques de Apartamentos identificados con Bloque A,B,C, cuenta con una población aproximada de 200 personas, tiene 3 parqueadero No:1 cuenta con 70 celdas, No:2 cuenta con 70 celdas, No:3 cuenta con 60 celdas, cada parqueadero tiene una entrada independiente y una salida para un total de 3 entradas y 3 salidas.

Requiere un sistema de visión Artificial para la seguridad del conjunto y todos sus habitantes, que registre todos sus usuarios y visitantes habilitados en el sistema, y que controle el fluido vehicular autónomo, al momento de que llegue un usuario del conjunto el proceda a verificar si el automóvil está registrado en el sistema le habrá la puerta automáticamente, de lo contrario deberá hacer el proceso pertinente de registro.

IMAGEN DEL CONDOMINIO



PROTOTIPO VISION ARTIFICIAL

CLASES DEL PROTOTIPO

CLASE PERSONA

CLASE APARTAMENTO

CLASE AUTOMOVIL

CLASE COLOR

CLASE MARCA

CLASE MODELO

CLASE PARQUEADERO

CLASE CELDA

CLASE CAMARAS

CLASE HISTORIAL

CLASE PERSONA

```
class Persona:
    def __init__(self, nombre = None, telefono = None, email = None, estado = None, tipo = None):

        self.nombre = nombre
        self.telefono = telefono
        self.email = email
        self.estado = estado
        self.tipo = tipo

    def getNombre(self):
```

```
    return self.nombre

def setNombre(self, nombre):
    self.nombre = nombre

def getTelefono(self):
    return self.telefono

def setTelefono(self, telefono):
    self.telefono = telefono

def getEmail(self):
    return self.email

def setEmail(self, email):
    self.email = email

def getEstado(self):
    return self.estado

def setEstado(self, estado):
    self.estado = estado

def getTipo(self):
    return self.tipo
    def setTipo(self, tipo):
    self.tipo = tipo
```

CLASE APARTAMENTO

```
class Apartamento:
    def __init__(self, Bloque = None, numero = None, id_persona = None):
        self.bloque = Bloque
        self.numero = numero
        self.id_persona = id_persona

    def get_Bloque(self):
        return self.Bloque

    def set_Bloque(self, Bloque):
        self.Bloque = Bloque

    def get_Numero(self):
        return self.numero

    def set_Numero(self, numero):
        self.numero = numero

    def get_id_persona(self):
        return self.id_persona

    def set_id_persona(self, id_persona):
        self.id_persona = id_persona
```

CLASE AUTOMOVIL

```
class Automovil:

    def __init__(self,placa = None, id_marca = None, id_modelo = None, id_color = None, estado = None,
id_persona = None):

        self.placa = placa

        self.id_marca = id_marca

        self.id_modelo = id_modelo

        self.id_color = id_color

        self.estado = estado

        self.id_persona = id_persona

    def get_Placa(self):

        return self.placa


    def set_Placa(self, placa):

        self.placa = placa


    def get_Marca(self):

        return self.id_marca


    def get_Modelo(self):

        return self.id_modelo


    def get_Color(self):

        return self.id_color


    def get_Estado(self):

        return self.estado
```

```
    def set_Estado(self, estado):
        self.estado = estado

    def get_id_persona(self):
        return self.id_persona
```

CLASE COLOR

```
class Color:
    def __init__(self, color = None):
        self.color = color

    def get_Color(self):
        return self.color

    def set_Color(self, color):
        self.color = color
```

CLASE MARCA

```
class Marca:
    def __init__(self, marca = None):

        self.marca = marca
    def get_Marca(self):
        return self.marca

    def set_Marca(self, marca):
        self.marca = marca
```

CLASE MODELO

```
class Modelo:
    def __init__(self,  modelo = None):

        self.modelo = modelo

    def get_Modelo(self):
        return self.modelo

    def set_Modelo(self, modelo):
        self.modelo = modelo
```

CLASE PARQUEADERO

```
class Parqueadero:
    def __init__(self, numero = None, totalceldas = None):
        self.numero = numero
        self.totalceldas = totalceldas

    def get_Numero(self):
        return self.numero

    def set_Numero(self, Placa):
        self.numero = numero

    def get_totalceldas(self):
        return self.totalceldas
```

```
    def set_totalceldas(self, totalceldas):
        self.totalceldas = totalceldas
```

CLASE CELDA

```
class Celda:
    def __init__(self, id_apartamento = None,id_parqueadero = None, numerocelda = None):

        self.id_apartamento = id_apartamento
        self.id_parqueadero = id_parqueadero
        self.numerocelda = numerocelda

    def get_id_apartamento(self):
        return self.id_apartamento

    def set_id_apartamento(self, id_apartamento):
        self.id_apartamento = id_apartamento

    def get_id_parqueadero(self):
        return self.id_parqueadero

    def set_id_parqueadero(self, id_parqueadero):
        self.id_parqueadero = id_parqueadero

    def get_numerocelda(self):
        return self.numerocelda

    def set_numerocelda(self, numerocelda):
        self.numerocelda = numerocelda
```

CLASE CAMARAS

```
class Camaras:
    def __init__(self, id_parqueadero = None ,numero = None, area = None, longitud = None):
        self.id_parqueadero = id_parqueadero
        self.numero = numero
        self.area = area
        self.longitud = longitud

    def get_id_parquedero(self):
        return self.id_parqueadero

    def set_id_parqueadero(self, id_parqueadero):
        self.id_parqueadero = id_parqueadero

    def get_Numero(self):
        return self.numero

    def set_Numero(self, Placa):
        self.numero = numero

    def get_area(self):
        return self.area

    def set_area(self, area):
        self.area = area

    def get_longitud(self):
```

```
        return self.longitud


    def set_longitud(self, longitud):
        self.longitud = longitud
```

CLASE HISTORIAL

```
class Historial:
    def __init__(self, id_automovil = None,id_camara = None ,fechahora = None):

        self.id_automovil = id_automovil
        self.id_camara=id_camara
        self.fechahora = fechahora


    def get_id_automovil(self):
        return self.id_automovil


    def set_id_automovil(self, id_automovil):
        self.id_automovil = id_automovil


    def get_id_camara(self):
        return self.id_camara


    def set_id_camara(self, id_camara):
        self.id_camara = id_camara


    def get_fechahora(self):
        return self.fechahora


    def set_fechahora(self, fechahora):
```

self.fechahora = fechahora

MODELO RELACIONAL VISION ARTIFICIAL

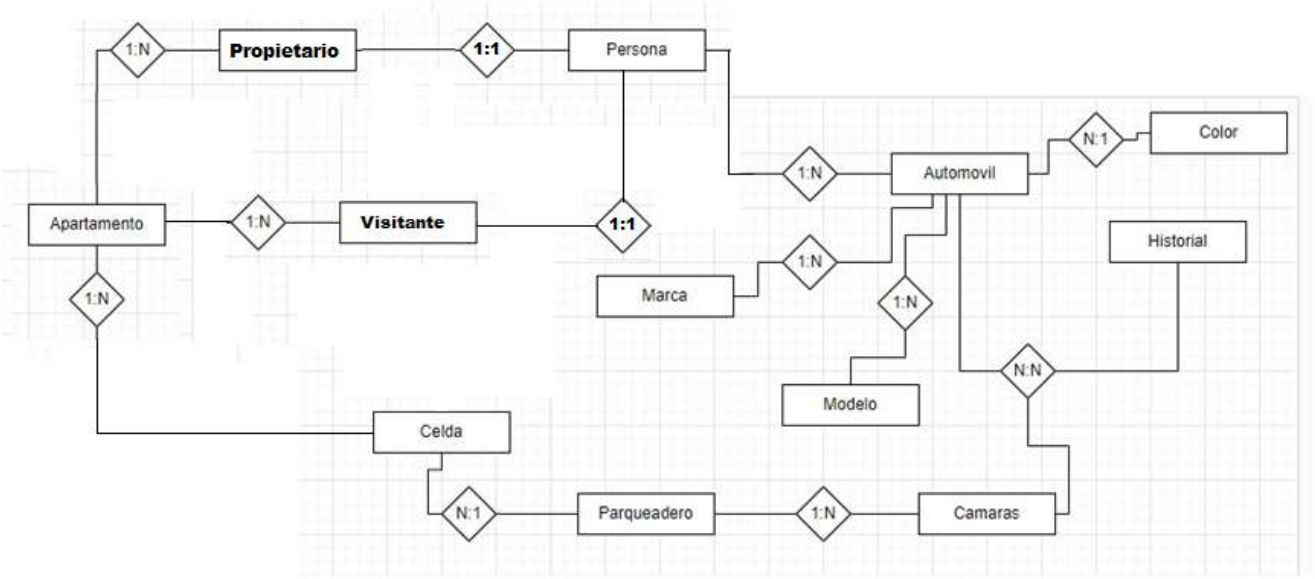
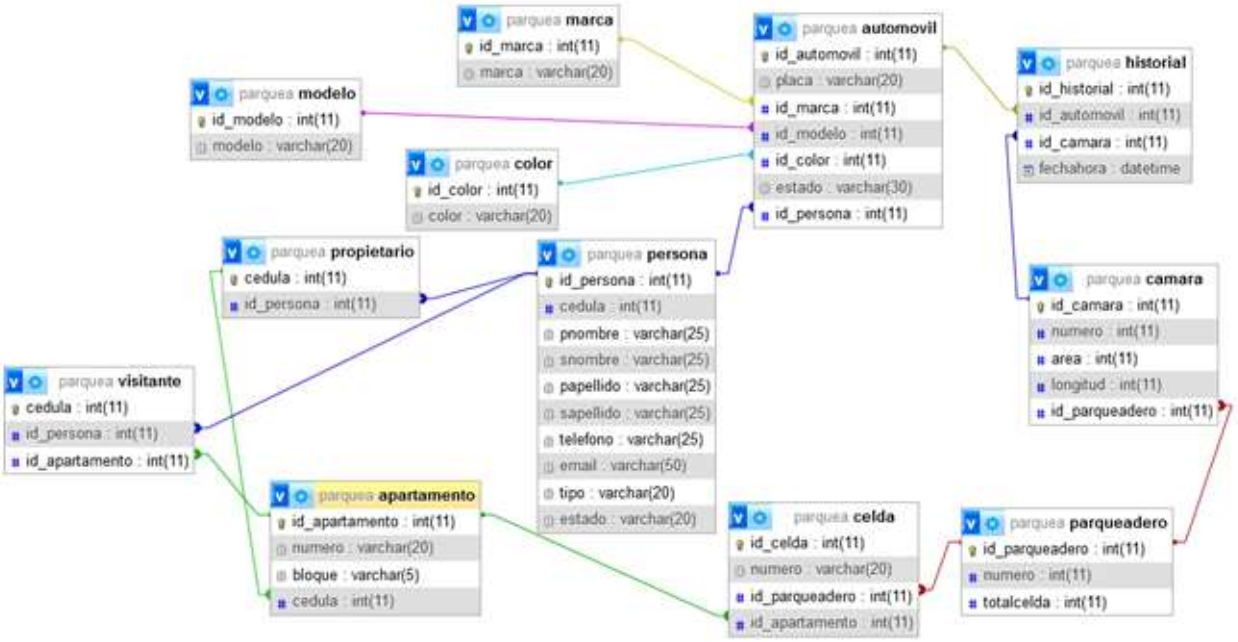


DIAGRAMA ENTIDAD RELACION VISION ARTIFICIAL



SCRIPTS DE LA BASE DE DATOS PARQUEADERO DEL PROTOTIPO VISION ARTIFICIAL

CREATE TABLE persona (

id_persona INTEGER PRIMARY KEY AUTO_INCREMENT,

nombre VARCHAR(35) NOT NULL,

telefono VARCHAR(35) NOT NULL,

email VARCHAR(35) NOT NULL,

estado INTEGER NOT NULL,

tipo VARCHAR(35) NOT NULL

```
);
CREATE TABLE popietario (
cedula INTEGER PRIMARY KEY ,
id_persona INTEGER NOT NULL,
FOREIGN KEY(id_persona) REFERENCES persona(id_persona));

CREATE TABLE visitante (
cedula INTEGER PRIMARY KEY ,
id_persona INTEGER NOT NULL,
id_apartamento INTEGER NOT NULL,
FOREIGN KEY(id_persona) REFERENCES persona(id_persona),
FOREIGN KEY(id_apartamento) REFERENCES persona(id_apartamento));

CREATE TABLE auto (
id_automovil INTEGER PRIMARY KEY AUTO_INCREMENT,
placa VARCHAR NOT NULL,
id_marca INTEGER NOT NULL,
id_modelo INTEGER NOT NULL,
id_color INTEGER NOT NULL,
estado INTEGER,
id_persona INTEGER,
FOREIGN KEY(id_marca) REFERENCES marca(id_marca),
FOREIGN KEY(id_modelo) REFERENCES modelo(id_modelo),
FOREIGN KEY(id_color) REFERENCES marca(id_color),
FOREIGN KEY(id_persona) REFERENCES persona(id_persona));

CREATE TABLE marca (
id_marca INTEGER PRIMARY KEY AUTO_INCREMENT,
marca VARCHAR(35) NOT NULL);
```

```
CREATE TABLE modelo (
id_modelo INTEGER PRIMARY KEY AUTO_INCREMENT,
modelo VARCHAR(35) NOT NULL);

CREATE TABLE color (
id_color INTEGER PRIMARY KEY AUTO_INCREMENT,
color VARCHAR(35) NOT NULL);


CREATE TABLE apartamento (
id_apartamento INTEGER PRIMARY KEY AUTO_INCREMENT,
bloque VARCHAR(35) NOT NULL,
numero INTEGER,
id_persona INTEGER NOT NULL,
FOREIGN KEY(id_persona) REFERENCES persona(id_persona));

CREATE TABLE parqueadero (
id_parqueadero INTEGER PRIMARY KEY AUTO_INCREMENT,
numero INTEGER NOT NULL,
totalceldas INTEGER);


CREATE TABLE celda (
id_celda INTEGER PRIMARY KEY AUTO_INCREMENT,
id_parqueadero INTEGER NOT NULL,
id_apartamento INTEGER NOT NULL,
numerocelda INTEGER NOT NULL,
FOREIGN KEY(id_parqueadero) REFERENCES parqueadero(id_parqueadero),
```

```
FOREIGN KEY(id_apartamento) REFERENCES apartamento(id_apartamento));

CREATE TABLE camaras (
id_camara INTEGER PRIMARY KEY AUTO_INCREMENT,
id_parqueadero INTEGER NOT NULL,
numero INTEGER NOT NULL,
area INTEGER NOT NULL,
longitud INTEGER NOT NULL,
FOREIGN KEY(id_parqueadero) REFERENCES parqueadero(id_parqueadero));

CREATE TABLE historial (
id_historial INTEGER PRIMARY KEY AUTO_INCREMENT,
id_automovil INTEGER,
id_camara INTEGER,
fechahora DATETIME,
FOREIGN KEY(id_automovil) REFERENCES auto(id_automovil),
FOREIGN KEY(id_camara) REFERENCES camaras(id_camara));
```

CONEXIÓN DE LA BASE DE DATOS PARQUEADERO DEL PROTOTIPO VISION ARTIFICIAL

```
import mysql.connector
class Conexion:
        def ObtenerConexion():
          conexion = mysql.connector.connect(user='root',
                                password='',
                                host='localhost',
                                database='parqueadero')
```

```
        return conexión
```

CRUD DEL PROTOTIPO VISION ARTIFICIAL

CLASE PADRE_D

```
from conexion import Conexion

class Padre_D:
    def insertar(INSERTAR,dato):
        with Conexion.ObtenerConexion() as conexion:
            cursor= conexion.cursor()
            cursor.execute(INSERTAR,dato)
            conexion.commit()
            print("el registro se guardo")
            conexion.close()
    def modificar(MODIFICAR,dato):
        with Conexion.ObtenerConexion() as conexion:
            cursor= conexion.cursor()
            cursor.execute(MODIFICAR,dato)
            print("el registro se actualizo")
    def eliminar(ELIMINAR,dato):
        with Conexion.ObtenerConexion() as conexion:
            cursor= conexion.cursor()
            cursor.execute(ELIMINAR,dato)
            print("el registro se ELIMINO")
    def consultar(SELECCIONAR):
        with Conexion.ObtenerConexion() as conexion:
            cursor= conexion.cursor()
            cursor.execute(SELECCIONAR)
            dato=cursor.fetchall()
        return dato
```

CLASE PERSONA_D

```
import sys
sys.path.append("../Modelo")
from Personas import Persona
from Apartamento import Apartamento
from propietario import Propietario
from Padre import Padre_D
class Persona_D(Padre_D):

    def guardar_persona(perso):
        INSERTAR='INSERT INTO persona VALUES(null,%s,%s,%s,%s,%s,%s,%s,%s,%s)'
        dato=(perso.cedula,perso.pnombre,perso.snombre,perso.papellido,perso.sapellido,
        perso.telefono,perso.email,perso.tipo,perso.estado)
        Persona_D.insertar(INSERTAR,dato)

    def guardar_propietario(perso):
        INSERTAR='INSERT INTO propietario VALUES(%s,%s)'
        dato=(perso.cedula,perso.id_persona)
        Persona_D.insertar(INSERTAR,dato)

    def guardar_visitante(perso):
        INSERTAR='INSERT INTO visitante VALUES(%s,%s,%s)'
        dato=(perso.cedula,perso.id_persona,perso.id_apartamento)
        Persona_D.insertar(INSERTAR,dato)

    def Modificar_persona(perso):
        MODIFICAR="UPDATE persona SET
        pnombre=%s,snombre=%s,papellido=%s,sapellido=%s,telefono=%s,
```

```
            email=%s where cedula=%s"

            dato=(perso.pnombre,perso.snombre,
            perso.papellido,perso.sapellido,perso.telefono,perso.telefono,perso.cedula)

            Persona_D.modificar(MODIFICAR,dato)


def validacedula(perso):

            SELECCIONAR='SELECT * FROM persona where cedula=%s'

            dato=(perso.cedula,)

            datos=Persona_D.consultardos(SELECCIONAR,dato)

            return datos


def mostrarid_persona(pro):

            SELECCIONAR='SELECT id_persona FROM persona where cedula=%s'

            dato=(pro.cedula,)

            datos=Persona_D.consultardos(SELECCIONAR,dato)

            return datos


def mostrarapartamento():

            SELECCIONAR='SELECT numero FROM apartamento '

            dato=Persona_D.consultar(SELECCIONAR)

            return dato


def mostrar_id_apa(apart):

            SELECCIONAR='SELECT id_apartamento FROM apartamento  where numero=%s and
            bloque=%s'

            dato=(apart.numero,apart.bloque)

            datos=Persona_D.consultardos(SELECCIONAR,dato)

            return datos
```

CLASE AUTOMOVIL_D

```
import sys
sys.path.append("../Modelo")
from Automovil import Automovil
from Padre import Padre_D
class Automovil_D(Padre_D):
    def guardarauto(auto):
        INSERTAR='INSERT INTO automovil VALUES(null,%s,%s,%s,%s,%s,%s)'
        dato=(auto.placa,auto.id_marca,auto.id_modelo,auto.id_color,auto.estado,auto.id_persona)
        Automovil_D.insertar(INSERTAR,dato)

    def mostrarmarca():
        SELECCIONAR='SELECT marca FROM marca '
        dato=Automovil_D.consultar(SELECCIONAR)
        return dato

    def consultamarca(mar):
        SELECCIONAR='SELECT id_marca FROM marca where marca=%s'
        data=(mar.marca,)
        dato=Automovil_D.consultardos(SELECCIONAR,data)
        return dato

    def mostrarmodelo():
        SELECCIONAR='SELECT modelo FROM modelo '
        dato=Automovil_D.consultar(SELECCIONAR)
        return dato

    def consultamodelo(mode):
```

```
        SELECCIONAR='SELECT id_modelo FROM modelo where modelo=%s'
        data=(mode.modelo,)
        dato=Automovil_D.consultardos(SELECCIONAR,data)
        return dato


def mostrarcolor():
        SELECCIONAR='SELECT color FROM color '
        dato=Automovil_D.consultar(SELECCIONAR)
        return dato

def consultacolor(colo):
        SELECCIONAR='SELECT id_color FROM color where color=%s'
        data=(colo.color,)
        dato=Automovil_D.consultardos(SELECCIONAR,data)
        return dato

def mostrarcedula():
        SELECCIONAR='SELECT cedula FROM persona '
        dato=Automovil_D.consultar(SELECCIONAR)
        return dato
def validaplaca(dato):
        SELECCIONAR='SELECT * FROM automovil where placa=%s'
        datos=Automovil_D.consultardos(SELECCIONAR,dato)
        return datos
def mostrarnombre(perso):
        dato=(perso.cedula,)
        SELECCIONAR='SELECT pnombre, snombre, papellido, sapellido,id_persona FROM persona
        where cedula=%s '
```

```
        datos=Automovil_D.consultardos(SELECCIONAR,dato)
        return datos
```

CLASE MARCA_D

CLASE MODELO_D

CLASE COLOR_D

```
import sys
sys.path.append("../Modelo")
from Color import Color
from Padre import Padre_D
class Color_D(Padre_D):
    def guardarcolor(col):
        INSERTAR='INSERT INTO color VALUES(null,%s)'
        dato=(col.color,)
        Color_D.insertar(INSERTAR,dato)
```

CLASE APARTAMENTO_D

```
import sys
sys.path.append("../Modelo")
from Automovil import Automovil
from Padre import Padre_D
class Apartamento_D(Padre_D):
    def guardarapartamento(apar):
        INSERTAR='INSERT INTO apartamento VALUES(null,%s,%s,%s)'
        dato=(apar.numero,apar.bloque,apar.cedula)
        Apartamento_D.insertar(INSERTAR,dato)
```

```
def mostrarcedula():
        SELECCIONAR='SELECT cedula FROM persona '
        dato=Apartamento_D.consultar(SELECCIONAR)
        return dato

def mostrarnombre(perso):
        dato=(perso.cedula,)
        SELECCIONAR='SELECT pnombre, snombre, papellido, sapellido FROM persona  where
        cedula=%s '
        datos=Apartamento_D.consultardos(SELECCIONAR,dato)
        return datos
```

VENTANAS DEL PROTOTIPO VISION ARTIFICIAL

EJECUCION DEL PROTOTIPO

```
import tkinter as tk
from tkinter import ttk
from Ventana_Principal import Ventana_Principal
from Ventana_automovil import Ventana_Automovil
def main():
        ventana_principal = Ventana_Principal()
        ventana_principal.mainloop()
if __name__ == '__main__':
        main()
```

VENTANA MENU PRINCIPAL




```
from tkinter import *
import tkinter as tk
from Ventana_Persona import Ventana_Persona
from Ventana_Automovil import Ventana_Automovil
from Ventana_Marca import Ventana_Marca
from Ventana_Modelo import Ventana_Modelo
from Ventana_Color import Ventana_Color
from tkinter import ttk
class Ventana_Principal(tk.Tk):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.config(width=1100, height=600)
        self.title("Vision Artificial")
        menuframe=Frame(self,bg="black")
```

```
menuframe.place(x= 0, y=0, width= 1100)
persona=Menubutton(menuframe,
                   text="Persona",
                   bg="white",
                   activeforeground='black',
                   activebackground='gray52')
automovil=Menubutton(menuframe,
                   text="Automovil",
                   bg="white",
                   activeforeground='black',
                   activebackground='gray52')
historial=Menubutton(menuframe,
                   text="Historial",
                   bg="white",
                   activeforeground='black',
                   activebackground='gray52')
apartamento=Menubutton(menuframe,
                   text="Apartamento",
                   bg="white",
                   activeforeground='black',
                   activebackground='gray52')
camaras=Menubutton(menuframe,
                   text="Camaras",
                   bg="white",
                   activeforeground='black',
                   activebackground='gray52')
parqueadero=Menubutton(menuframe,
                   text="Parqueadero",
                   bg="white",
                   activeforeground='black',
```

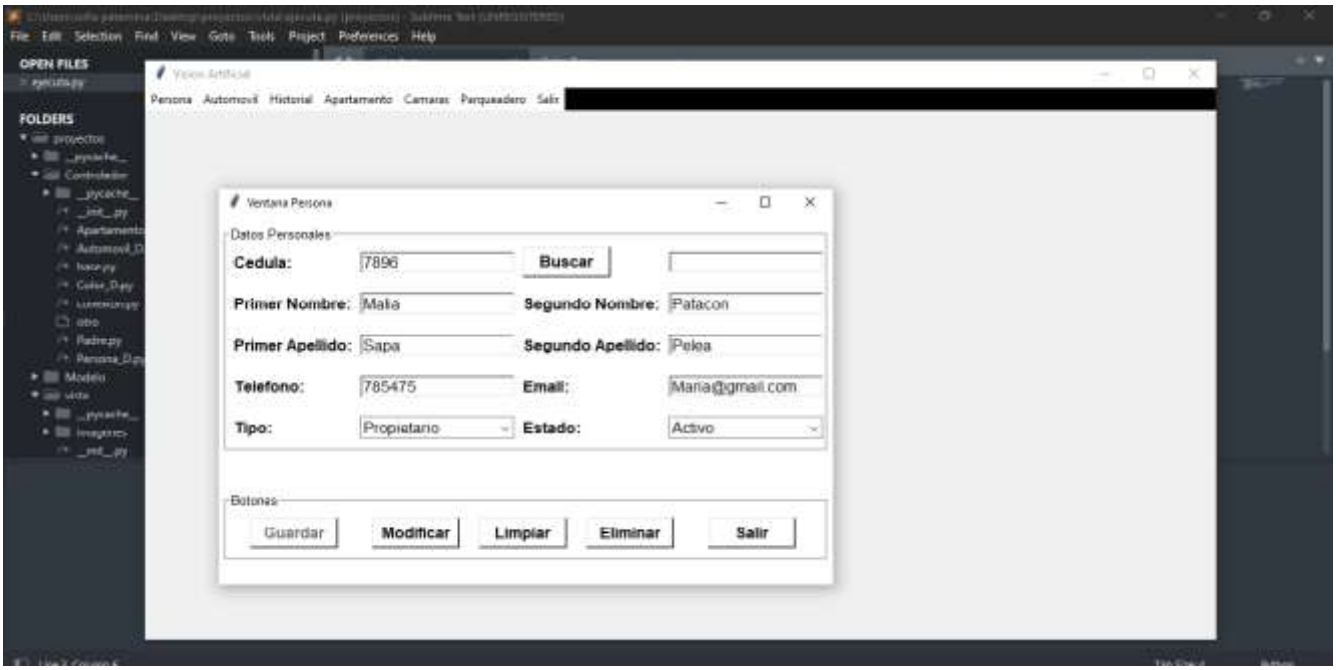
```
                activebackground='gray52')
salir=Menubutton(menuframe,
                text="Salir",
                bg="white",
                activeforeground='black',
                activebackground='gray52')
menu_persona=Menu(persona,tearoff=0)
menu_persona.add_command(label="Registro",
                  command=Ventana_Persona,
                  background="black",
                  foreground="white",
                  activeforeground="black",
                  activebackground="gray52")
menu_automovil=Menu(automovil,tearoff=0)
menu_automovil.add_command(label="Registro",
                  command=Ventana_Automovil,
                  background="black",
                  foreground="white",
                  activeforeground="black",
                  activebackground="gray52")
menu_automovil.add_command(label="Marca",
                  command=Ventana_Marca,
                  background="black",
                  foreground="white",
                  activeforeground="black",
                  activebackground="gray52")
menu_automovil.add_command(label="Modelo",
                  command=Ventana_Modelo,
                  background="black",
                  foreground="white",
```

```
                activeforeground="black",
                activebackground="gray52")
menu_automovil.add_command(label="Color",
                command=Ventana_Color,
                background="black",
                foreground="white",
                activeforeground="black",
                activebackground="gray52")
menu_historial=Menu(historial,tearoff=0)
menu_historial.add_command(label="Historial",
                command=lambda:print("hola"),
                background="black",
                foreground="white",
                activeforeground="black",
                activebackground="gray52")
menu_apartamento=Menu(apartamento,tearoff=0)
menu_apartamento.add_command(label="Apartamento",
                command=lambda:print("hola"),
                background="black",
                foreground="white",
                activeforeground="black",
                activebackground="gray52")
menu_camaras=Menu(camaras,tearoff=0)
menu_camaras.add_command(label="Camaras",
                command=lambda:print("hola"),
                background="black",
                foreground="white",
                activeforeground="black",
                activebackground="gray52")
menu_parqueadero=Menu(parqueadero,tearoff=0)
```

```
menu_parqueadero.add_command(label="Parqueadero",
                command=lambda:print("hola"),
                background="black",
                foreground="white",
                activeforeground="black",
                activebackground="gray52")
menu_parqueadero.add_command(label="Celda",
                command=lambda:print("hola"),
                background="black",
                foreground="white",
                activeforeground="black",
                activebackground="gray52")
menu_salir=Menu(salir,tearoff=0)
menu_salir.add_command(label="Salir",
                command=lambda:self.destroy(),
                background="black",
                foreground="white",
                activeforeground="black",
                activebackground="gray52")
persona.config(menu=menu_persona)
persona.pack(side="left")
automovil.config(menu=menu_automovil)
automovil.pack(side="left")
historial.config(menu=menu_historial)
historial.pack(side="left")
apartamento.config(menu=menu_apartamento)
apartamento.pack(side="left")
camaras.config(menu=menu_camaras)
camaras.pack(side="left")
parqueadero.config(menu=menu_parqueadero)
```

```
parqueadero.pack(side="left")
salir.config(menu=menu_salir)
salir.pack(side="left")
```

VENTANA PERSONA




```
import sys
sys.path.append("../Controlador")
sys.path.append("../Modelo")
import tkinter as tk
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
from PIL import Image
from PIL import ImageTk
import imutils
from Personas import Persona
from propietario import Propietario
from visitante import Visitante
from Persona_D import Persona_D
```

```
from Apartamento import Apartamento
class Ventana_Persona(tk.Toplevel):
   def __init__(self, *args, **kwargs):
      super().__init__(*args, **kwargs)
      self.config(background="white")
      self.title("Ventana Persona")
      self.geometry("630x400")
      #******************************variables de las entradas*************
      self.pnombre1=StringVar()
      self.snombre1=StringVar()
      self.papellido1=StringVar()
      self.sapellido1=StringVar()
      self.telefono1=StringVar()
      self.email1=StringVar()
      self.tipo1=StringVar()
      self.estado1=StringVar()
      self.eid_persona1=StringVar()
      self.eid_apartamento=StringVar()


      #****************************************** Frame
Label**********************************
      self.lframe_persona = LabelFrame(self,text="Datos Personales",font=("Roboto
Cn",10),bg="white",fg="black")
      self.lframe_persona.place(x= 5, y=10, width= 620)
      #****************************************Frame
Botones**********************************
      self.lframe_boton = LabelFrame(self,text="Botones",font=("Roboto Cn",10),bg="white",fg="black")
      self.lframe_boton.place(x= 5, y=300, width= 620)


      #*************************************imagenes de los
botones**********************************
```

```
        """
        self.imguardar=Image.open("imagenes/guardar.png")
        self.imguardar = self.imguardar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imguardar=ImageTk.PhotoImage(self.imguardar)

        self.imeditar=Image.open("imagenes/modificar.png")
        self.imeditar = self.imeditar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imeditar=ImageTk.PhotoImage(self.imeditar)

        self.imeliminar=Image.open("imagenes/Eliminar.png")
        self.imeliminar = self.imeliminar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imeliminar=ImageTk.PhotoImage(self.imeliminar)

        self.imsalir=Image.open("imagenes/salir.png")
        self.imsalir = self.imsalir.resize((60, 60), Image.Resampling.LANCZOS)
        self.imsalir=ImageTk.PhotoImage(self.imsalir)

        self.imbuscar=Image.open("imagenes/buscar.png")
        self.imbuscar = self.imbuscar.resize((40, 40), Image.Resampling.LANCZOS)
        self.imbuscar=ImageTk.PhotoImage(self.imbuscar)
        """
        #**********************************label*******************************************
        self.lcedula=Label(self.lframe_persona,text="Cedula:",font=("Roboto Cn",12,"bold"),bg="white")
        self.lcedula.grid(row=0,column=0,pady=10,padx=5,sticky="w")

        self.lpnombre=Label(self.lframe_persona,text="Primer Nombre:",font=("Roboto
Cn",12,"bold"),bg="white")
        self.lpnombre.grid(row=1,column=0,pady=10,padx=5,sticky="w")

        self.lsnombre=Label(self.lframe_persona,text="Segundo Nombre:",font=("Roboto
Cn",12,"bold"),bg="white")
```

```
        self.lsnombre.grid(row=1,column=2,pady=10,padx=5,sticky="w")


        self.lpapellido=Label(self.lframe_persona,text="Primer Apellido:",font=("Roboto
Cn",12,"bold"),bg="white")
        self.lpapellido.grid(row=2,column=0,pady=10,padx=5,sticky="w")


        self.lsapellido=Label(self.lframe_persona,text="Segundo Apellido:",font=("Roboto
Cn",12,"bold"),bg="white")
        self.lsapellido.grid(row=2,column=2,pady=10,padx=5,sticky="w")


        self.ltelefono=Label(self.lframe_persona,text="Telefono:",font=("Roboto Cn",12,"bold"),bg="white")
        self.ltelefono.grid(row=3,column=0,pady=10,padx=5,sticky="w")


        self.lemail=Label(self.lframe_persona,text="Email:",font=("Roboto Cn",12,"bold"),bg="white")
        self.lemail.grid(row=3,column=2,pady=10,padx=5,sticky="w")


        self.ltipo=Label(self.lframe_persona,text="Tipo:",font=("Roboto Cn",12,"bold"),bg="white")
        self.ltipo.grid(row=4,column=0,pady=10,padx=5,sticky="w")


        self.lestado=Label(self.lframe_persona,text="Estado:",font=("Roboto Cn",12,"bold"),bg="white")
        self.lestado.grid(row=4,column=2,pady=10,padx=5,sticky="w")


        self.lapartamento=Label(self.lframe_persona,text="Apartamento:",font=("Roboto
Cn",12,"bold"),bg="white")
        self.lapartamento.grid(row=5,column=0,pady=10,padx=5,sticky="w")


        self.lbloque=Label(self.lframe_persona,text="Bloque:",font=("Roboto Cn",12,"bold"),bg="white")
        self.lbloque.grid(row=5,column=2,pady=10,padx=5,sticky="w")
#*******************************Entradas************************************
        self.ecedula=Entry(self.lframe_persona,font=("Roboto Cn",12),width=17,bd=2)
        self.ecedula.grid(row=0,column=1,pady=10)
```

```
self.eid_persona=Entry(self.lframe_persona,textvariable=self.eid_persona1,font=("Roboto
Cn",12),width=17,bd=2)

self.eid_persona.grid(row=0,column=3,pady=10)



self.epnombre=Entry(self.lframe_persona,textvariable=self.pnombre1,font=("Roboto
Cn",12),width=17,bd=2)

self.epnombre.grid(row=1,column=1,pady=10)


self.esnombre=Entry(self.lframe_persona,textvariable=self.snombre1,font=("Roboto
Cn",12),width=17,bd=2)

self.esnombre.grid(row=1,column=3,pady=10)


self.epapellido=Entry(self.lframe_persona,textvariable=self.papellido1,font=("Roboto
Cn",12),width=17,bd=2)

self.epapellido.grid(row=2,column=1,pady=10)


self.esapellido=Entry(self.lframe_persona,textvariable=self.sapellido1,font=("Roboto
Cn",12),width=17,bd=2)

self.esapellido.grid(row=2,column=3,pady=10)


self.etelefono=Entry(self.lframe_persona,textvariable=self.telefono1,font=("Roboto
Cn",12),width=17,bd=2)

self.etelefono.grid(row=3,column=1,pady=10)


self.eemail=Entry(self.lframe_persona,textvariable=self.email1,font=("Roboto Cn",12),width=17,bd=2)

self.eemail.grid(row=3,column=3,pady=10)


self.etipo=ttk.Combobox(self.lframe_persona,textvariable=self.tipo1,font=("Roboto Cn",12),width=15)

self.etipo.grid(row=4,column=1, pady=10)

self.etipo["value"]=["Propietario","Visitante"]
```

```
        self.etipo.bind("<<ComboboxSelected>>",self.ejecutapartamento)


        self.eestado=ttk.Combobox(self.lframe_persona,textvariable=self.estado1,font=("Roboto
Cn",12),width=15)
        self.eestado.grid(row=4,column=3, pady=10)
        self.eestado["value"]=["Activo","inactivo"]


        dato=Persona_D.mostrarapartamento()
        self.eapartamento=ttk.Combobox(self.lframe_persona,font=("Roboto Cn",12),width=15)
        self.eapartamento.grid(row=5,column=1,pady=10)
        self.eapartamento["value"]=dato


        self.ebloque=ttk.Combobox(self.lframe_persona,font=("Roboto Cn",12),width=15)
        self.ebloque.grid(row=5,column=3,pady=10)
        self.ebloque["value"]=["A","B","C"]


        self.lapartamento.grid_forget()
        self.eapartamento.grid_forget()
        self.lbloque.grid_forget()
        self.ebloque.grid_forget()


#**********************************botones*************************************
        self.bguardar=Button(self.lframe_boton,width=8,text="Guardar",font=("Roboto
Cn",12,"bold"),command=self.guardarpersona,background="white")
        self.bguardar.grid(row=5,column=0,pady=10,padx=25)
        self.bmodificar=Button(self.lframe_boton,width=8,text="Modificar",font=("Roboto
Cn",12,"bold"),background="white",command=self.Modificar_persona)
        self.bmodificar.grid(row=5,column=1,pady=10,padx=10)
        self.belimpiar=Button(self.lframe_boton,width=8,text="Limpiar",font=("Roboto
Cn",12,"bold"),background="white",command=self.limpiar)
        self.belimpiar.grid(row=5,column=2,pady=10,padx=10)
```

```
        self.beliminar=Button(self.lframe_boton,width=8,text="Eliminar",font=("Roboto
Cn",12,"bold"),background="white")

        self.beliminar.grid(row=5,column=3,pady=10,padx=10)

        self.bsalir=Button(self.lframe_boton,width=8,text="Salir",command=self.destroy,font=("Roboto
Cn",12,"bold"),background="white")

        self.bsalir.grid(row=5,column=4,pady=10,padx=25)

        self.buscar=Button(self.lframe_persona,width=8,text="Buscar",font=("Roboto
Cn",12,"bold"),background="white",command=self.validacedula)

        self.buscar.place(x=305,y=5)


        self.epnombre.config(state=DISABLED)

        self.esnombre.config(state=DISABLED)

        self.epapellido.config(state=DISABLED)

        self.esapellido.config(state=DISABLED)

        self.etelefono.config(state=DISABLED)

        self.eemail.config(state=DISABLED)

        self.etipo.config(state=DISABLED)

        self.eestado.config(state=DISABLED)

        self.bguardar.config(state=DISABLED)

        self.bmodificar.config(state=DISABLED)

        self.beliminar.config(state=DISABLED)



        #self.grab_set()
    #*************************************************************************
    def ejecutapartamento(self,event=None):
        if self.etipo.get()=="Propietario":
            self.lapartamento.grid_forget()
            self.eapartamento.grid_forget()
            self.lbloque.grid_forget()
            self.ebloque.grid_forget()
```

```
        if self.etipo.get()=="Visitante":
            self.lapartamento.grid(row=5,column=0,pady=10,padx=5,sticky="w")
            self.eapartamento.grid(row=5,column=1,pady=10)
            self.lbloque.grid(row=5,column=2,pady=10,padx=5,sticky="w")
            self.ebloque.grid(row=5,column=3,pady=10)


#DESACTIVA LOS LABEL, LAS ENTRADAS Y LOS BOTONES DE LA VENTANA PERSONA


    #**************************limpia las entradas**************************
    def limpiar(self):
        self.ecedula.delete(0,END)
        self.epnombre.delete(0,END)
        self.esnombre.delete(0,END)
        self.epapellido.delete(0,END)
        self.esapellido.delete(0,END)
        self.etelefono.delete(0,END)
        self.eemail.delete(0,END)
        self.eestado.delete(0,END)
        self.etipo.delete(0,END)
        self.ecedula.focus_set()
    #**********************************validar cedula************************

    def validacedula(self):
        a=self.ecedula.get()
        if (len(a)==0):
            messagebox.showinfo(message="¿Digite la cedula porfavor?", title="validar campo")
            self.ecedula.focus_set()
        if a.isnumeric()==False:
```

```
            messagebox.showinfo(message="¿Digite valores numericos porfavor?", title="validar campo")
            self.ecedula.delete(0,END)
            self.ecedula.focus_set()


        else:
            perso=Persona(cedula=self.ecedula.get())
            personal=Persona_D.validacedula(perso)
            if personal==[]:
                messagebox.showinfo(message="¿la cedula no esta registrada en el sistema?", title="Validar
cedula")
                self.epnombre.config(state=NORMAL)
                self.esnombre.config(state=NORMAL)
                self.epapellido.config(state=NORMAL)
                self.esapellido.config(state=NORMAL)
                self.etelefono.config(state=NORMAL)
                self.eemail.config(state=NORMAL)
                self.etipo.config(state=NORMAL)
                self.eestado.config(state=NORMAL)
                self.bguardar.config(state=NORMAL)
                self.epnombre.focus_set()
            else:


                for i in personal:
                    self.pnombre1.set(i[2])
                    self.snombre1.set(i[3])
                    self.papellido1.set(i[4])
                    self.sapellido1.set(i[5])
                    self.telefono1.set(i[6])
                    self.email1.set(i[7])
                    self.tipo1.set(i[8])
```

```
            self.estado1.set(i[9])


            messagebox.showinfo(message="¿la cedula  esta registrada en el sistema?", title="Validar cedula")
            self.bguardar.config(state=DISABLED)
            self.bmodificar.config(state=NORMAL)
            self.beliminar.config(state=NORMAL)
            self.epnombre.config(state=NORMAL)
            self.esnombre.config(state=NORMAL)
            self.epapellido.config(state=NORMAL)
            self.esapellido.config(state=NORMAL)
            self.etelefono.config(state=NORMAL)
            self.eemail.config(state=NORMAL)
            self.etipo.config(state=NORMAL)
            self.eestado.config(state=NORMAL)
            self.epnombre.focus_set()


    #**********************************funcion guardar persona ***************
    def guardarpersona(self):
        a=set(self.ecedula.get())
        if (len(a)==0):
            messagebox.showinfo(message="¿Digite la cedula  porfavor?", title="validar cedula")
            self.ecedula.focus_set()
        else:
            m=messagebox.askquestion(message="¿Desea Guardar el registro?", title="Guardar")
            if m=="yes":

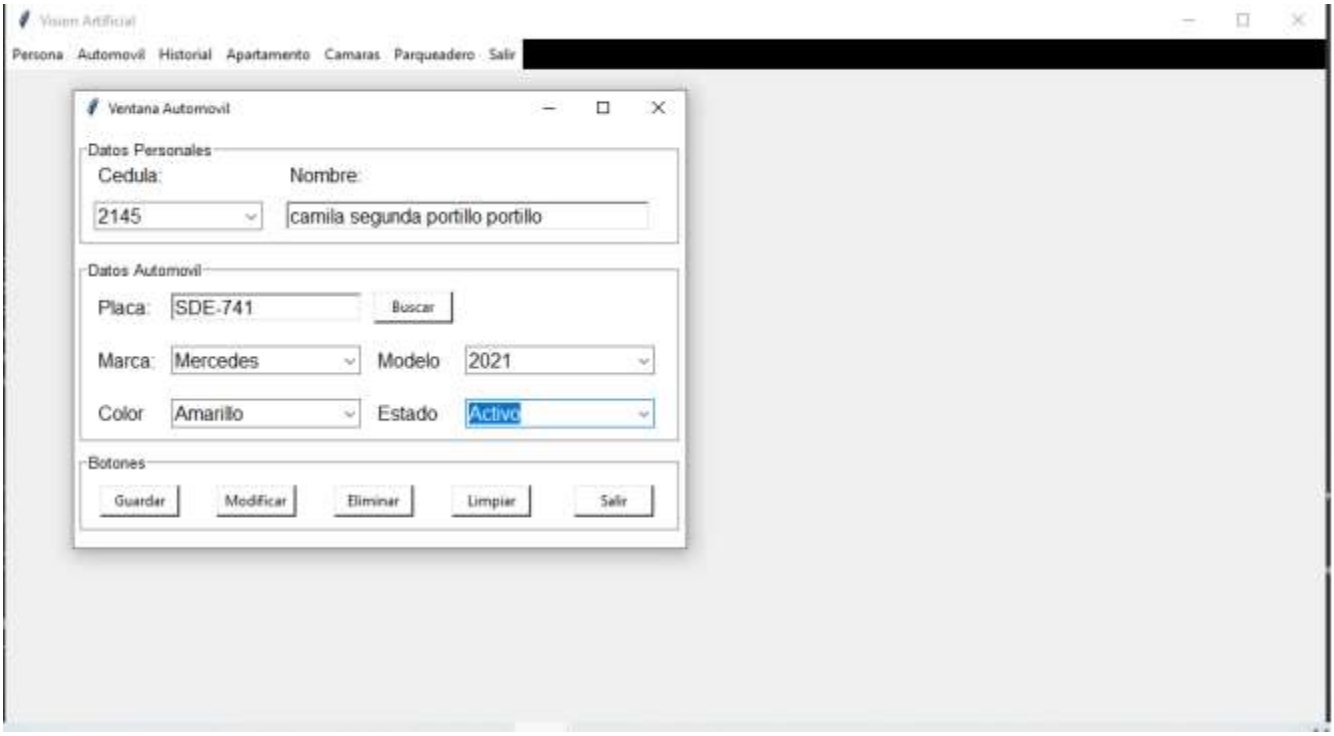
perso=Persona(self.ecedula.get(),self.epnombre.get(),self.esnombre.get(),self.epapellido.get(),self.esapellid
o.get(),self.etelefono.get(),self.eemail.get(),self.etipo.get(),self.eestado.get())
            Persona_D.guardar_persona(perso)
            pro=Persona(cedula=self.ecedula.get())
            x=Persona_D.mostrarid_persona(pro)
```

```
            if self.etipo.get()=="Propietario":
                prop=Propietario(cedula=self.ecedula.get(),id_persona=x[0][0])
                Persona_D.guardar_propietario(prop)
            else:
                visi=Apartamento(numero=self.eapartamento.get(),bloque=self.ebloque.get())
                y=Persona_D.mostrar_id_apa(visi)
                prov=Visitante(cedula=self.ecedula.get(),id_persona=x[0][0],id_apartamento=y[0][0])
                Persona_D.guardar_visitante(prov)
            messagebox.showinfo(message="¿Se guardo correctamente el registro?", title="Guardar persona")
            self.limpiar
        else:
            messagebox.showinfo(message="¿El registro no se guardo?", title="Guardar persona")
    def Modificar_persona(self):
        m=messagebox.askquestion(message="¿Desea Guardar el registro?", title="Guardar")
        if m=="yes":

perso=Persona(pnombre=self.epnombre.get(),snombre=self.esnombre.get(),papellido=self.epapellido.get(),s
apellido=self.esapellido.get(),telefono=self.etelefono.get(),email=self.eemail.get(),tipo=self.etipo.get(),estad
o=self.eestado.get(),cedula=self.ecedula.get())
            Persona_D.Modificar_persona(perso)
            messagebox.showinfo(message="¿Se Modifico correctamente el registro?", title="Modificar persona")
            self.limpiar
        else:
            messagebox.showinfo(message="¿El registro no se Modifico?", title="Modificar persona")
```


www.sena.edu.co
SENAComunica

VENTANA AUTOMOVIL




```
import sys
import tkinter as tk
from tkinter import *
from tkinter import ttk
from PIL import Image
from PIL import ImageTk
import imutils
from tkinter import messagebox
sys.path.append("../Controlador")
sys.path.append("../Modelo")
from Personas import Persona
from Automovil import Automovil
from Modelo import Modelo
from Marca import Marca
```

```
from Color import Color
from Automovil_D import Automovil_D
class Ventana_Automovil(tk.Toplevel):
   def __init__(self, *args, **kwargs):
      super().__init__(*args, **kwargs)
      global texto
      texto=StringVar()
      #********************configurar ventana**********************
      self.config(background="white")
      self.title("Ventana Automovil")
      self.geometry("510x350")

      #***************** Frame Label***********************************
      self.lframe_persona = LabelFrame(self,text="Datos Personales",font=("Roboto
Cn",10),bg="white",fg="black")
      self.lframe_persona.place(x= 5, y=10, width= 500)
      #********************* Frame Label***********************************
      self.lframe_automovil = LabelFrame(self,text="Datos Automovil",font=("Roboto
Cn",10),bg="white",fg="black")
      self.lframe_automovil.place(x= 5, y=110, width= 500)
      #******************Frame Botones***********************************
      self.lframe_boton = LabelFrame(self,text="Botones",font=("Roboto Cn",10),bg="white",fg="black")
      self.lframe_boton.place(x= 5, y=270, width= 500)

      self.lcedula=Label(self.lframe_persona,text="Cedula:",font=("Roboto Cn",12),bg="white")
      self.lcedula.grid(row=0,column=0,padx=10,sticky="w")

      self.lnombre=Label(self.lframe_persona,text="Nombre:",font=("Roboto Cn",12),bg="white")
      self.lnombre.grid(row=0,column=1,padx=10,sticky="w")
```

```
        self.lplaca=Label(self.lframe_automovil,text="Placa:",font=("Roboto Cn",12),bg="white")
        self.lplaca.grid(row=0,column=0,pady=10,padx=10,sticky="w")

        self.lidmarca=Label(self.lframe_automovil,text="Marca:",font=("Roboto Cn",12),bg="white")
        self.lidmarca.grid(row=1,column=0,pady=10,padx=10,sticky="w")


        self.lidmodelo=Label(self.lframe_automovil,text="Modelo",font=("Roboto Cn",12),bg="white")
        self.lidmodelo.grid(row=1,column=2,pady=10,padx=10,sticky="w")


        self.lidcolor=Label(self.lframe_automovil,text="Color",font=("Roboto Cn",12),bg="white")
        self.lidcolor.grid(row=2,column=0,pady=10,padx=10,sticky="w")

        self.lestado=Label(self.lframe_automovil,text="Estado",font=("Roboto Cn",12),bg="white")
        self.lestado.grid(row=2,column=2,pady=10,padx=10,sticky="w")

    #*********************************Entradas*************************************


        dato=Automovil_D.mostrarcedula()
        self.ecedula=ttk.Combobox(self.lframe_persona,width=13,font=("Roboto Cn",12))
        self.ecedula.grid(row=1,column=0,pady=10,padx=10)
        self.ecedula["value"]=dato
        self.ecedula.current(0)
        self.ecedula.bind("<<ComboboxSelected>>",self.ejecutanombre)

        self.enombre=Entry(self.lframe_persona,textvariable=texto,font=("Roboto Cn",12),width=33,bd=2)
        self.enombre.grid(row=1,column=1,pady=10,padx=10)
```

```
self.eplaca=Entry(self.lframe_automovil,font=("Roboto Cn",12),width=17,bd=2)
self.eplaca.grid(row=0,column=1)

madato=Automovil_D.mostrarmarca()
self.eidmarca=ttk.Combobox(self.lframe_automovil,font=("Roboto Cn",12),width=15)
self.eidmarca.grid(row=1,column=1)
self.eidmarca["value"]=madato
self.eidmarca.current(0)
self.eidmarca.bind("<<ComboboxSelected>>",self.ejecutamarca)

modato=Automovil_D.mostrarmodelo()
self.eidmodelo=ttk.Combobox(self.lframe_automovil,font=("Roboto Cn",12),width=15)
self.eidmodelo.grid(row=1,column=3)
self.eidmodelo["value"]=modato
self.eidmodelo.current(0)
self.eidmodelo.bind("<<ComboboxSelected>>",self.ejecutamodelo)

mcolor=Automovil_D.mostrarcolor()
self.eidcolor=ttk.Combobox(self.lframe_automovil,font=("Roboto Cn",12),width=15)
self.eidcolor.grid(row=2,column=1)
self.eidcolor["value"]=mcolor
self.eidcolor.current(0)
self.eidcolor.bind("<<ComboboxSelected>>",self.ejecutacolor)

self.eestado=ttk.Combobox(self.lframe_automovil,font=("Roboto Cn",12),width=15)
self.eestado.grid(row=2,column=3)
self.eestado["value"]=["Activo","Inactivo"]
#*************************************************************************
"""
self.imguardar=Image.open("imagenes/guardar.png")
```

```
        self.imguardar = self.imguardar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imguardar=ImageTk.PhotoImage(self.imguardar)

        self.imeditar=Image.open("imagenes/modificar.png")
        self.imeditar = self.imeditar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imeditar=ImageTk.PhotoImage(self.imeditar)

        self.imeliminar=Image.open("imagenes/eliminar.png")
        self.imeliminar = self.imeliminar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imeliminar=ImageTk.PhotoImage(self.imeliminar)

        self.imsalir=Image.open("imagenes/salir.png")
        self.imsalir = self.imsalir.resize((80, 30), Image.Resampling.LANCZOS)
        self.imsalir=ImageTk.PhotoImage(self.imsalir)

        self.imbuscar=Image.open("imagenes/buscar.png")
        self.imbuscar = self.imbuscar.resize((40, 40), Image.Resampling.LANCZOS)
        self.imbuscar=ImageTk.PhotoImage(self.imbuscar)
        """
#**********************************Botones*************************************
        self.buscar=Button(self.lframe_automovil,text="Buscar",bg="white",width=8)
        self.buscar.grid(row=0,column=2,padx=10)

self.bguardar=Button(self.lframe_boton,text="Guardar",bg="white",command=self.guardarautomovil,width
=8)
        self.bguardar.grid(row=0,column=0,padx=15,pady=10)
        self.bmodificar=Button(self.lframe_boton,text="Modificar",bg="white",width=8)
        self.bmodificar.grid(row=0,column=1,padx=16,pady=10)
        self.beliminar=Button(self.lframe_boton,text="Eliminar",bg="white",width=8)
        self.beliminar.grid(row=0,column=2,padx=16,pady=10)
        self.blimpiar=Button(self.lframe_boton,text="Limpiar",bg="white",width=8)
```

```
        self.blimpiar.grid(row=0,column=3,padx=16,pady=10)
        self.bsalir=Button(self.lframe_boton,text="Salir",bg="white",width=8,command=self.destroy)
        self.bsalir.grid(row=0,column=4,padx=20,pady=10)


    def ejecutanombre(self,event=None):
        global texto
        persona1=Persona(cedula=self.ecedula.get())
        persona=Automovil_D.mostrarnombre(persona1)
        for i in persona:
            texto.set(i[0]+" "+i[1]+" "+i[2]+" "+i[3])
            self.id_persona=i[4]
        self.eplaca.focus_set()
#**************************Marca**************************************
    def ejecutamarca(self,event=None):
        mar=Marca(marca=self.eidmarca.get())
        self.marca=Automovil_D.consultamarca(mar)
#**************************Modelo*************************************
    def ejecutamodelo(self,event=None):
        mode=Modelo(modelo=self.eidmodelo.get())
        self.modelo=Automovil_D.consultamodelo(mode)
#**************************Color**************************************
    def ejecutacolor(self,event=None):
        colo=Color(color=self.eidcolor.get())
        self.color=Automovil_D.consultacolor(colo)
#*********************************Guarda Automovil*********************************
    def guardarautomovil(self):
        pla=self.eplaca.get()
        if self.eplaca.get()==" ":
            messagebox.showinfo(message="¿Digite el nombre  porfavor?", title="validar placa")
            self.eplaca.focus_set()
```

else:

m=messagebox.askquestion(message="¿Desea Guardar el registro?", title="Guardar")

if m=="yes":

auto=Automovil(self.eplaca.get(),self.marca[0][0],self.modelo[0][0],self.color[0][0],self.eestado.get(),self.id_persona)

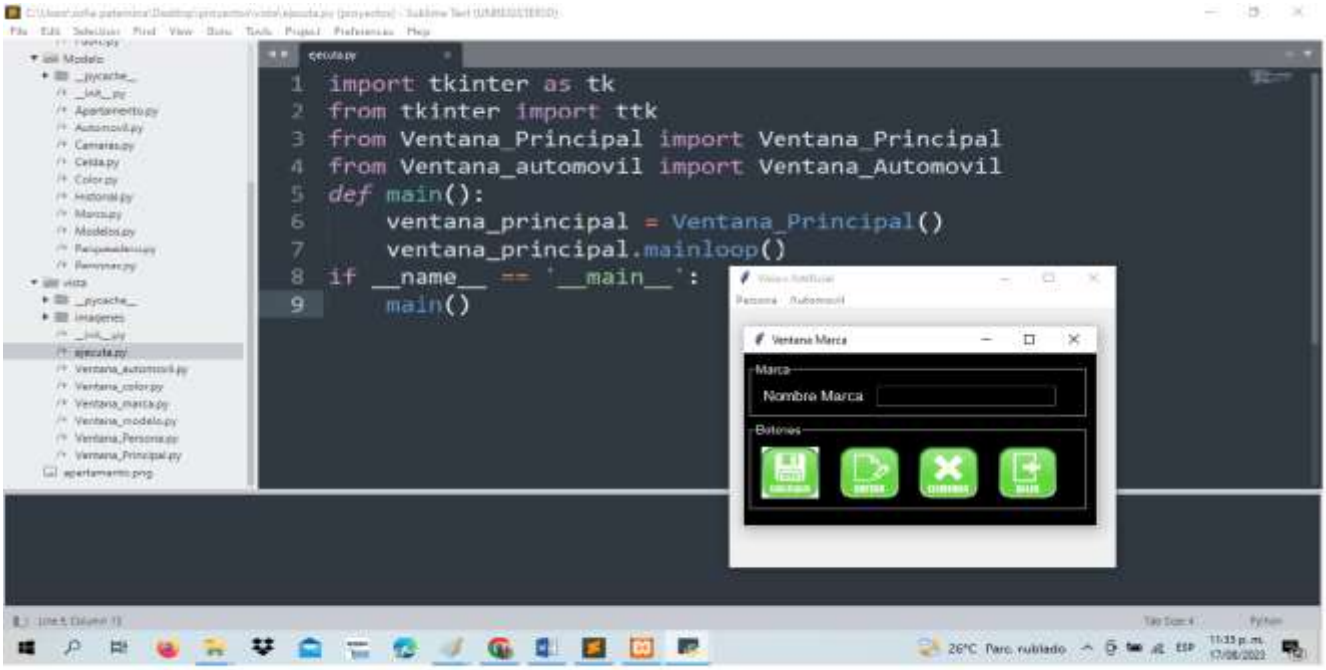
Automovil_D.guardarauto(auto)

messagebox.showinfo(message="¿Se guardo correctamente el registro?", title="Guardar Automovil")

else:

messagebox.showinfo(message="¿El registro no se guardo?", title="Guardar Automovil")

VENTANA MARCA



import sys

import tkinter as tk

from tkinter import *

from tkinter import ttk

from tkinter import messagebox

from PIL import Image

```
from PIL import ImageTk
import imutils
sys.path.append("../Controlador")
sys.path.append("../Modelo")
from conexion import Conexion
from Marca import Marca
from Hijos_D import Marca_D
class Ventana_Marca(tk.Toplevel):
    """Clase Ventana_Marca """
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        #****************Configuracion Ventana ***************************
        self.config(background="black")
        self.title("Ventana Marca")
        self.geometry("365x200")

        #************** Frame Label***********************
        self.lframe_marca = LabelFrame(self,text="Marca",font=("Roboto Cn",10),bg="black",fg="white")
        self.lframe_marca.place(x= 5, y=10, width= 350)
        #*******************Frame Botones***********************
        self.lframe_boton = LabelFrame(self,text="Botones",font=("Roboto Cn",10),bg="black",fg="white")
        self.lframe_boton.place(x= 5, y=80, width= 350)
        #**************label *****************************************
        self.lmarca=Label(self.lframe_marca,text="Nombre Marca",font=("Roboto
Cn",12),bg="black",fg="white")
        self.lmarca.grid(row=0,column=0,padx=10,pady=10)
        #*********************Entrada ***************************
        self.emarca=Entry(self.lframe_marca,font=("Roboto Cn",12),bg="black",fg="white",bd=2)
        self.emarca.grid(row=0,column=1)
        #*************************** Imagenes  de los Botones ******************************
```

```
        self.imguardar=Image.open("imagenes/guardar.png")
        self.imguardar = self.imguardar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imguardar=ImageTk.PhotoImage(self.imguardar)

        self.imeditar=Image.open("imagenes/editar.png")
        self.imeditar = self.imeditar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imeditar=ImageTk.PhotoImage(self.imeditar)

        self.imeliminar=Image.open("imagenes/eliminar.png")
        self.imeliminar = self.imeliminar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imeliminar=ImageTk.PhotoImage(self.imeliminar)

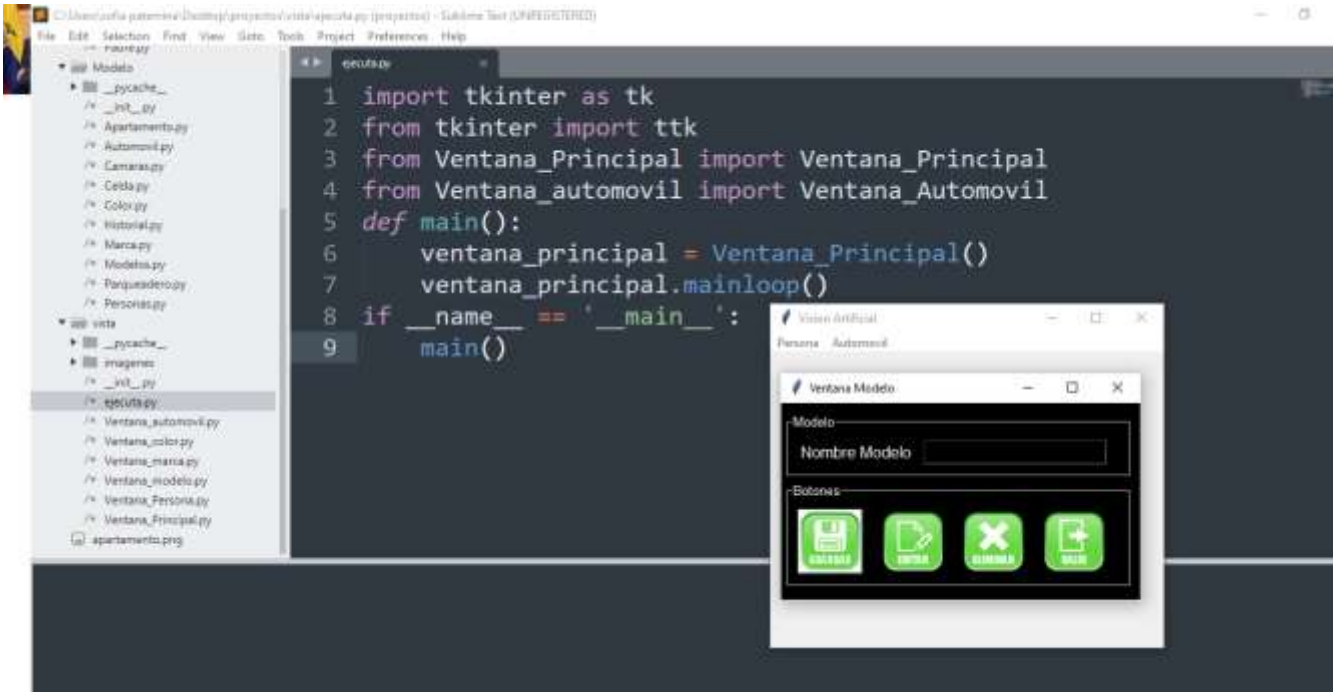
        self.imsalir=Image.open("imagenes/salir.png")
        self.imsalir = self.imsalir.resize((60, 60), Image.Resampling.LANCZOS)
        self.imsalir=ImageTk.PhotoImage(self.imsalir)

        self.imbuscar=Image.open("imagenes/buscar.png")
        self.imbuscar = self.imbuscar.resize((40, 40), Image.Resampling.LANCZOS)
        self.imbuscar=ImageTk.PhotoImage(self.imbuscar)
        #************************ Botones ***************************************

self.bguardar=Button(self.lframe_boton,image=self.imguardar,bg="black",bd="0",command=self.guardarmarca)
        self.bguardar.grid(row=5,column=0,pady=10,padx=10)
        self.bmodificar=Button(self.lframe_boton,image=self.imeditar,bg="black",bd="0")
        self.bmodificar.grid(row=5,column=1,pady=10,padx=10)
        self.beliminar=Button(self.lframe_boton,image=self.imeliminar,bg="black",bd="0")
        self.beliminar.grid(row=5,column=2,pady=10,padx=10)
        self.bsalir=Button(self.lframe_boton,image=self.imsalir,bg="black",bd="0",command=self.destroy)
        self.bsalir.grid(row=5,column=3,pady=10,padx=10)
        self.focus()
```

```
    self.grab_set()
    #****************** Metodo Guaradar ****************************
  def guardarmarca(self):
    if self.emarca.get()==" ":
      messagebox.showinfo(message="¿Digite el nombre de la Marca porfavor?", title="validar Marca")
      self.emarca.focus_set()
    else:
      m=messagebox.askquestion(message="¿Desea Guardar el registro?", title="Guardar")
      if m=="yes":
        marca=Marca(self.emarca.get(),)
        INSERTAR='INSERT INTO marca VALUES(null,%s)'
        dato=[marca.marca]
        Marca_D.insertar(INSERTAR,dato)
        messagebox.showinfo(message="¿Se guardo correctamente el registro?", title="Guardar Marca")
      else:
        messagebox.showinfo(message="¿El registro no se guardo?", title="Guardar color")
```

VENTANA MODELO



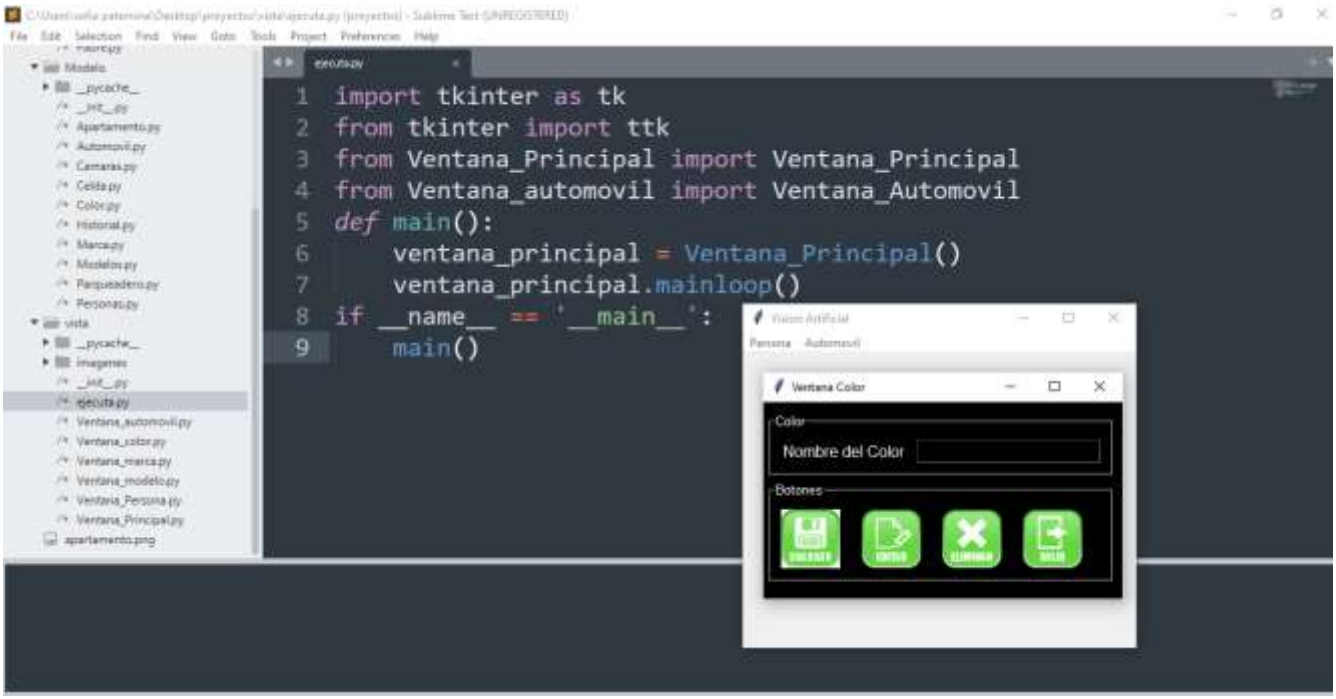

```
import sys
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
sys.path.append("../Controlador")
sys.path.append("../Modelo")
from conexion import Conexion
from Personas import Persona
from Hijos_D import Persona_D
class Ventana_Persona(tk.Toplevel):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.config(width=300, height=200)
        self.title("Ventana secundaria")
        self.lnombre=ttk.Label(self,text="Nombre",font=("Roboto Cn",12))
        self.lnombre.grid(row=0,column=0,pady=5,padx=5,sticky="w")
        self.lnombre.config(background="white")
        self.ltelefono=ttk.Label(self,text="Telefono",font=("Roboto Cn",12))
```

```
        self.ltelefono.grid(row=1,column=0,pady=5,padx=5,sticky="w")
        self.ltelefono.config(background="white")
        self.lemail=ttk.Label(self,text="Email",font=("Roboto Cn",12))
        self.lemail.grid(row=2,column=0,pady=5,padx=5,sticky="w")
        self.lemail.config(background="white")
        self.lestado=ttk.Label(self,text="Estado",font=("Roboto Cn",12))
        self.lestado.grid(row=3,column=0,pady=5,padx=5,sticky="w")
        self.lestado.config(background="white")
        self.ltipo=ttk.Label(self,text="Tipo",font=("Roboto Cn",12))
        self.ltipo.grid(row=4,column=0,pady=5,padx=5,sticky="w")
        self.ltipo.config(background="white")
    #*********************************Entradas*************************************
        self.enombre=tk.Entry(self,font=("Roboto Cn",12),width=17,bd=2)
        self.enombre.grid(row=0,column=1)
        self.eemail=tk.Entry(self,font=("Roboto Cn",12),width=17,bd=2)
        self.eemail.grid(row=1,column=1)
        self.etelefono=tk.Entry(self,font=("Roboto Cn",12),width=17,bd=2)
        self.etelefono.grid(row=2,column=1)
        self.etipo=ttk.Combobox(self,font=("Roboto Cn",12),width=15)
        self.etipo.grid(row=3,column=1)
        self.etipo["value"]=["Activo","Inactivo"]
        self.eestado=ttk.Combobox(self,font=("Roboto Cn",12),width=15)
        self.eestado.grid(row=4,column=1)
        self.eestado["value"]=["Propietario","Arrendatario","Visitante"]
    #*********************************botones*************************************
        self.bguardar=ttk.Button(self,text="Guardar",command=self.guardarpersona)
        self.bguardar.grid(row=5,column=0,pady=10,padx=30)
        self.bmodificar=ttk.Button(self,text="Modificar")
        self.bmodificar.grid(row=5,column=1,pady=10,padx=30)
        self.beliminar=ttk.Button(self,text="Eliminar")
```

```
        self.beliminar.grid(row=5,column=2,pady=10,padx=30)
        self.bsalir=ttk.Button(self,text="Salir")
        self.bsalir.grid(row=5,column=3,pady=10,padx=20)
        self.focus()
        self.grab_set()
    def guardarpersona(self):
        if self.enombre.get()==" ":
            messagebox.showinfo(message="¿Digite el nombre  porfavor?", title="validar nombre")
            self.enombre.focus_set()
        else:
            m=messagebox.askquestion(message="¿Desea Guardar el registro?", title="Guardar")
            if m=="yes":

perso=Persona(self.enombre.get(),self.etelefono.get(),self.eemail.get(),self.etipo.get(),self.eestado.get())
                INSERTAR='INSERT INTO persona VALUES(null,%s,%s,%s,%s,%s)'
                dato=[perso.nombre,perso.telefono,perso.email,perso.estado,perso.tipo]
                Persona_D.insertar(INSERTAR,dato)
                messagebox.showinfo(message="¿Se guardo correctamente el registro?", title="Guardar persona")
            else:
                messagebox.showinfo(message="¿El registro no se guardo?", title="Guardar persona")
```

VENTANA COLOR




```
import sys
import tkinter as tk
from tkinter import *
from tkinter import ttk
from PIL import Image
from PIL import ImageTk
import imutils
from tkinter import messagebox
sys.path.append("../Controlador")
sys.path.append("../Modelo")
from conexion import Conexion
from Color import Color
from Hijos_D import Color_D
class Ventana_Color(tk.Toplevel):
    """Clase Ventana_Color """
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
```

```
        #****************************************Configuracion Ventana ***************************
        self.config(background="black")
        self.title("Ventana Color")
        self.geometry("365x200")

        #********************** Frame Label***********************************
        self.lframe_color = LabelFrame(self,text="Color",font=("Roboto Cn",10),bg="black",fg="white")
        self.lframe_color.place(x= 5, y=10, width= 350)
        #********************Frame Botones***********************************
        self.lframe_boton = LabelFrame(self,text="Botones",font=("Roboto Cn",10),bg="black",fg="white")
        self.lframe_boton.place(x= 5, y=80, width= 350)
        #************************label ****************************************
        self.lcolor=Label(self.lframe_color,text="Nombre del Color",font=("Roboto
Cn",12),bg="black",fg="white")
        self.lcolor.grid(row=0,column=0,padx=10,pady=10)
        #******************Entrada ****************************************
        self.ecolor=Entry(self.lframe_color,font=("Roboto Cn",12),bg="black",fg="white",bd=2)
        self.ecolor.grid(row=0,column=1)
        #******************* Imagenes  de los Botones ******************************
        self.imguardar=Image.open("imagenes/guardar.png")
        self.imguardar = self.imguardar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imguardar=ImageTk.PhotoImage(self.imguardar)

        self.imeditar=Image.open("imagenes/editar.png")
        self.imeditar = self.imeditar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imeditar=ImageTk.PhotoImage(self.imeditar)

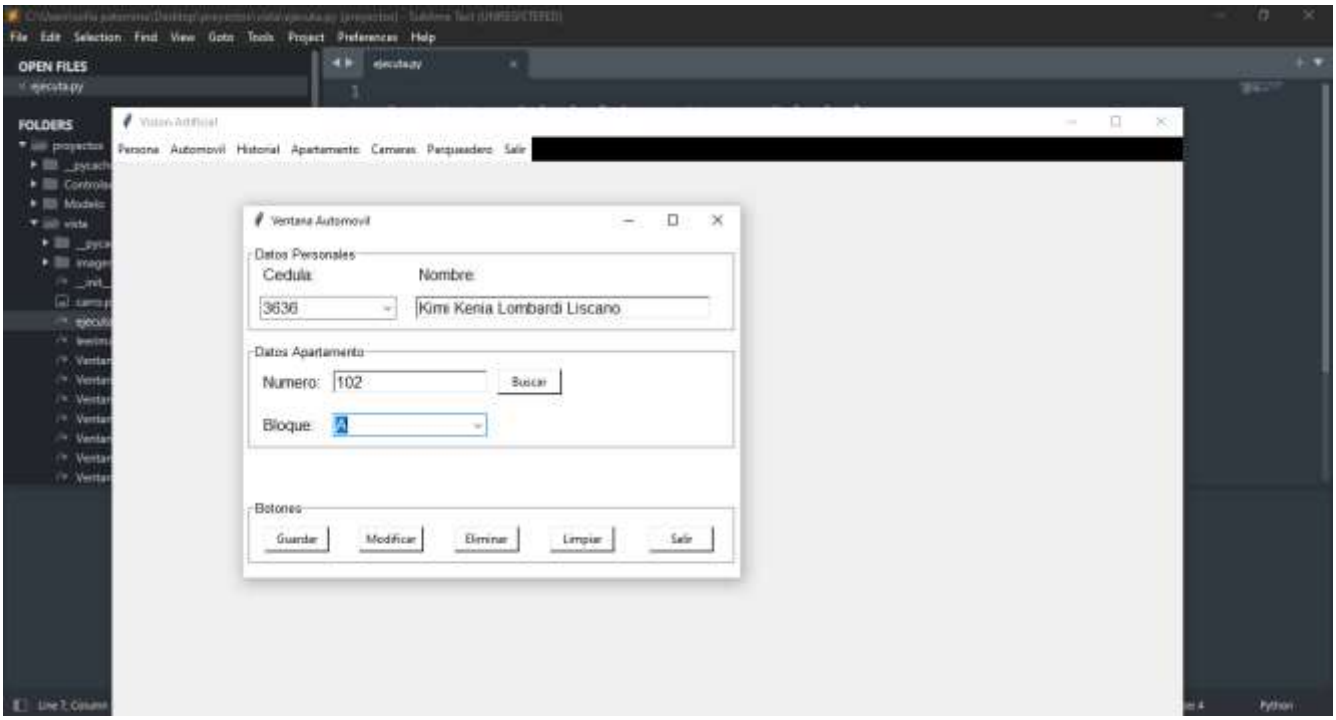
        self.imeliminar=Image.open("imagenes/eliminar.png")
        self.imeliminar = self.imeliminar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imeliminar=ImageTk.PhotoImage(self.imeliminar)
```

```
    self.imsalir=Image.open("imagenes/salir.png")
    self.imsalir = self.imsalir.resize((60, 60), Image.Resampling.LANCZOS)
    self.imsalir=ImageTk.PhotoImage(self.imsalir)

    self.imbuscar=Image.open("imagenes/buscar.png")
    self.imbuscar = self.imbuscar.resize((40, 40), Image.Resampling.LANCZOS)
    self.imbuscar=ImageTk.PhotoImage(self.imbuscar)
    #****************** Botones ******************************
    self.bguarda=Button(self.lframe_boton,image=self.imguardar,bg="black",bd="0")
    self.bguarda.grid(row=5,column=0,pady=10,padx=10)
    self.bmodificar=Button(self.lframe_boton,image=self.imeditar,bg="black",bd="0")
    self.bmodificar.grid(row=5,column=1,pady=10,padx=10)
    self.beliminar=Button(self.lframe_boton,image=self.imeliminar,bg="black",bd="0")
    self.beliminar.grid(row=5,column=2,pady=10,padx=10)
    self.bsalir=Button(self.lframe_boton,image=self.imsalir,bg="black",bd="0",command=self.destroy)
    self.bsalir.grid(row=5,column=3,pady=10,padx=10)
    self.focus()
    self.grab_set()
    #********************** Metodo Guaradar ********************
def guardarcolor(self):
    if self.ecolor.get()==" ":
        messagebox.showinfo(message="¿Digite el nombre del color porfavor?", title="validar color")
        self.ecolor.focus_set()
    else:
        m=messagebox.askquestion(message="¿Desea Guardar el registro?", title="Guardar")
        if m=="yes":
            color=Color(self.ecolor.get(),)
            INSERTAR='INSERT INTO color VALUES(null,%s)'
            dato=[color.color]
```

```
    Color_D.insertar(INSERTAR,dato)
    messagebox.showinfo(message="¿Se guardo correctamente el registro?", title="Guardar color")
else:
    messagebox.showinfo(message="¿El registro no se guardo?", title="Guardar color")
```

VENTANA APARTAMENTO




```
import sys
import tkinter as tk
from tkinter import *
from tkinter import ttk
from PIL import Image
from PIL import ImageTk
import imutils
from tkinter import messagebox
sys.path.append("../Controlador")
sys.path.append("../Modelo")
from Apartamento import Apartamento
```

```
from Personas import Persona
from Apartamento_D import Apartamento_D
class Ventana_Apartamento(tk.Toplevel):
   def __init__(self, *args, **kwargs):
      super().__init__(*args, **kwargs)
      global texto
      texto=StringVar()
      #**************************configurar ventana**********************
      self.config(background="white")
      self.title("Ventana Automovil")
      self.geometry("510x350")

      #************************** Frame Label*****************************
      self.lframe_persona = LabelFrame(self,text="Datos Personales",font=("Roboto
Cn",10),bg="white",fg="black")
      self.lframe_persona.place(x= 5, y=10, width= 500)
      #***************************** Frame Label************************
      self.lframe_apartamento = LabelFrame(self,text="Datos Apartamento",font=("Roboto
Cn",10),bg="white",fg="black")
      self.lframe_apartamento.place(x= 5, y=110, width= 500)
      #****************************************Frame
Botones***********************************
      self.lframe_boton = LabelFrame(self,text="Botones",font=("Roboto Cn",10),bg="white",fg="black")
      self.lframe_boton.place(x= 5, y=270, width= 500)

      self.lcedula=Label(self.lframe_persona,text="Cedula:",font=("Roboto Cn",12),bg="white")
      self.lcedula.grid(row=0,column=0,padx=10,sticky="w")

      self.lnombre=Label(self.lframe_persona,text="Nombre:",font=("Roboto Cn",12),bg="white")
      self.lnombre.grid(row=0,column=1,padx=10,sticky="w")
```

```
        self.lnumero=Label(self.lframe_apartamento,text="Numero:",font=("Roboto Cn",12),bg="white")
        self.lnumero.grid(row=0,column=0,pady=10,padx=10,sticky="w")

        self.lbloque=Label(self.lframe_apartamento,text="Bloque:",font=("Roboto Cn",12),bg="white")
        self.lbloque.grid(row=1,column=0,pady=10,padx=10,sticky="w")

#*******************************Entradas************************************


        dato=Apartamento_D.mostrarcedula()
        self.ecedula=ttk.Combobox(self.lframe_persona,width=13,font=("Roboto Cn",12))
        self.ecedula.grid(row=1,column=0,pady=10,padx=10)
        self.ecedula["value"]=dato
        self.ecedula.current(0)
        self.ecedula.bind("<<ComboboxSelected>>",self.ejecutanombre)

        self.enombre=Entry(self.lframe_persona,textvariable=texto,font=("Roboto Cn",12),width=33,bd=2)
        self.enombre.grid(row=1,column=1,pady=10,padx=10)

        self.enumero=Entry(self.lframe_apartamento,font=("Roboto Cn",12),width=17,bd=2)
        self.enumero.grid(row=0,column=1)


        self.ebloque=ttk.Combobox(self.lframe_apartamento,font=("Roboto Cn",12),width=15)
        self.ebloque.grid(row=1,column=1)
        self.ebloque["value"]=["A","B","C"]


        #*************************************************************************
```

```
        """
        self.imguardar=Image.open("imagenes/guardar.png")
        self.imguardar = self.imguardar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imguardar=ImageTk.PhotoImage(self.imguardar)

        self.imeditar=Image.open("imagenes/modificar.png")
        self.imeditar = self.imeditar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imeditar=ImageTk.PhotoImage(self.imeditar)

        self.imeliminar=Image.open("imagenes/eliminar.png")
        self.imeliminar = self.imeliminar.resize((60, 60), Image.Resampling.LANCZOS)
        self.imeliminar=ImageTk.PhotoImage(self.imeliminar)

        self.imsalir=Image.open("imagenes/salir.png")
        self.imsalir = self.imsalir.resize((80, 30), Image.Resampling.LANCZOS)
        self.imsalir=ImageTk.PhotoImage(self.imsalir)

        self.imbuscar=Image.open("imagenes/buscar.png")
        self.imbuscar = self.imbuscar.resize((40, 40), Image.Resampling.LANCZOS)
        self.imbuscar=ImageTk.PhotoImage(self.imbuscar)
        """
#*********************************Botones*************************************
        self.buscar=Button(self.lframe_apartamento,text="Buscar",bg="white",width=8)
        self.buscar.grid(row=0,column=2,padx=10)

self.bguardar=Button(self.lframe_boton,text="Guardar",bg="white",command=self.guardarapartamento,wi
dth=8)
        self.bguardar.grid(row=0,column=0,padx=15,pady=10)
        self.bmodificar=Button(self.lframe_boton,text="Modificar",bg="white",width=8)
        self.bmodificar.grid(row=0,column=1,padx=16,pady=10)
        self.beliminar=Button(self.lframe_boton,text="Eliminar",bg="white",width=8)
```

```
        self.beliminar.grid(row=0,column=2,padx=16,pady=10)
        self.blimpiar=Button(self.lframe_boton,text="Limpiar",bg="white",width=8)
        self.blimpiar.grid(row=0,column=3,padx=16,pady=10)
        self.bsalir=Button(self.lframe_boton,text="Salir",bg="white",width=8,command=self.destroy)
        self.bsalir.grid(row=0,column=4,padx=20,pady=10)


    def ejecutanombre(self,event=None):
        global texto
        persona1=Persona(cedula=self.ecedula.get())
        persona=Apartamento_D.mostrarnombre(persona1)
        for i in persona:
            texto.set(i[0]+" "+i[1]+" "+i[2]+" "+i[3])
        self.enumero.focus_set()


#*********************************Guarda Aartamento*******************************
    def guardarapartamento(self):


        if self.enumero.get()==" ":
            messagebox.showinfo(message="¿Digite el numero del apartamento  porfavor?", title="validar
apartamento")
            self.enumero.focus_set()
        else:
            m=messagebox.askquestion(message="¿Desea Guardar el registro?", title="Guardar")
            if m=="yes":
                apa=Apartamento(self.enumero.get(),self.ebloque.get(),self.ecedula.get())
                Apartamento_D.guardarapartamento(apa)
                messagebox.showinfo(message="¿Se guardo correctamente el registro?", title="Guardar
Automovil")
            else:
                messagebox.showinfo(message="¿El registro no se guardo?", title="Guardar Automovil")
```

PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- **Denominación del Programa de Formación:** Programación de Software.
- **Código del Programa de Formación:** 233104 V2
- **Nombre del Proyecto:** Desarrollo de software a la medida para el sector empresarial
- **Fase del Proyecto:** Evaluación
- **Actividad de Proyecto:** Verificar la funcionalidad del software de acuerdo a los requerimientos planteados en la documentación del proyecto formativo
- **Competencia:** 220501096 - Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo
- **Resultados de Aprendizaje Alcanzar:** 669168 - RAP 3. Codificar el software empleando el lenguaje de programación seleccionado.
- **Duración de la Guía:** 240 Horas

2. PRESENTACIÓN

Estimados aprendices, es un placer darle la bienvenida a esta guía sobre la codificación de software empleando el lenguaje de programación seleccionado. En este material, encontrarán una completa introducción que abarcará los aspectos fundamentales necesarios para dominar la escritura de código de manera efectiva y eficiente. A lo largo de este recorrido, nos sumergiremos en los conceptos clave y las mejores prácticas de programación, brindándoles ejemplos claros y desafiantes ejercicios que les permitirán poner en práctica sus conocimientos y fortalecer sus habilidades. Esta guía sentará las bases necesarias para que puedan embarcarse en su camino en el fascinante mundo de la codificación y dar vida a sus ideas a través del poder del software.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

En el mundo de la codificación, existen numerosos lenguajes de programación que se han convertido en pilares fundamentales en la industria del software.

En esta guía, exploraremos algunos de los lenguajes más populares y ampliamente adoptados, como Python, Java, C++, JavaScript y Php. Cada uno de estos lenguajes tiene sus propias características y aplicaciones, lo que los convierte en herramientas poderosas para abordar diferentes necesidades de programación. Ya sea que estén interesados en el desarrollo web, la inteligencia artificial, el desarrollo de aplicaciones móviles o la programación de sistemas, este recorrido les proporcionará una visión general de estos lenguajes y les ayudará a comprender sus conceptos básicos. Prepárense para sumergirse en el apasionante mundo de la programación y descubrir las posibilidades que estos lenguajes tienen para ofrecer.

3.1 Reflexión Inicial

Antes de comenzar nuestro recorrido por el mundo de la programación, es importante tomarnos un momento para reflexionar sobre el impacto que tiene en nuestras vidas. Piensa en todas las veces que has interactuado con un software, ya sea a través de una aplicación en tu teléfono móvil, un programa en tu computadora o incluso un dispositivo inteligente en tu hogar. ¿Te has preguntado alguna vez cómo es posible que estos programas funcionen y respondan a tus comandos?

Te invito a reflexionar sobre las siguientes preguntas:

- ¿Cuál es el papel de la programación en nuestra sociedad moderna? ¿En qué aspectos de nuestra vida diaria podemos identificar su influencia?
- ¿Por qué crees que es importante aprender a programar? ¿Cuáles crees que son los beneficios y las oportunidades que ofrece esta habilidad?
- ¿Qué te inspira a explorar el mundo de la programación? ¿Hay algún proyecto o idea que te gustaría desarrollar a través del software?

Tómate un momento para reflexionar sobre estas preguntas y anota tus pensamientos. Recuerda que la programación es una herramienta poderosa que puede cambiar la forma en que interactuamos con el mundo y nos brinda la capacidad de convertir nuestras ideas en realidad.

Duración: 2 Hora

3.2 Actividades de contextualización e identificación de conocimientos

Lenguajes de programación

- Enumera al menos cinco lenguajes de programación que conozcas.
- Junto a cada lenguaje de programación, escribe una breve descripción de su uso o aplicación principal.

Conceptos básicos de programación

Completa las siguientes afirmaciones:

- Una variable es...
- Un bucle "for" se utiliza para...
- La estructura "if-else" se utiliza para...

Programación orientada a objetos

- Explica brevemente qué es la programación orientada a objetos.
- Menciona al menos tres conceptos fundamentales de la programación orientada a objetos.
- ¿Cuál es la diferencia entre una clase y un objeto en la programación orientada a objetos?

Duracion: 2 Horas

3.3 Actividades de apropiación del conocimiento (conceptualización y teorización)

Es crucial resaltar la importancia de las buenas prácticas en el desarrollo y codificación de software. Las buenas prácticas son pautas y estándares que nos guían hacia la creación de código limpio, legible, eficiente y mantenible.

Estas prácticas nos ayudan a construir programas robustos y confiables, reducir errores y facilitar la colaboración entre desarrolladores. Al seguir las buenas prácticas, nos aseguramos de que nuestro código sea comprensible tanto para nosotros como para otros programadores, lo que facilita su mantenimiento y actualización a lo largo del tiempo.

Además, las buenas prácticas nos permiten optimizar el rendimiento de nuestras aplicaciones y garantizar la seguridad de los datos y la privacidad de los usuarios. En resumen, al implementar buenas prácticas en el desarrollo y codificación de software, estamos sentando las bases para construir soluciones de calidad que satisfagan las necesidades de los usuarios y contribuyan al éxito de nuestros proyectos.

3.3.1 Actividad de aprendizaje AA1 - Construir una aplicación web utilizando el lenguaje de programación seleccionado, siguiendo las mejores prácticas de codificación.

3.3.1.1 Planificar el proyecto: En base a la situación problema definida por el instructor, crear un plan detallado que incluye la definición del problema a resolver, los objetivos del proyecto, los requisitos funcionales y no funcionales, así como un esquema básico de la arquitectura del sistema. Luego, presenta tu plan al resto de la formación, compartiendo su enfoque y justificando las decisiones tomadas. Esta actividad fomenta el pensamiento crítico, la planificación y la habilidad para comunicar ideas de manera efectiva.

Evidencias:

Evidencia de producto: AA1-EV01 – Planificación del proyecto:

- Un plan detallado del proyecto que incluye la definición del problema, los objetivos, los requisitos funcionales y no funcionales, y un esquema de la arquitectura del sistema.
- Una presentación clara y concisa del plan, destacando las decisiones tomadas y justificando las elecciones realizadas.
- Participación activa en la discusión de los planes de otros grupos, aportando ideas y realizando preguntas relevantes.

3.3.1.2 Revisar y analizar código fuente: El instructor proporcionará fragmentos de código de una aplicación web y se le pide que los revise en busca de posibles mejoras y errores. Aplica las mejores prácticas de codificación aprendidas, como utilizar una nomenclatura adecuada, agregar comentarios descriptivos y utilizar estructuras de control eficientes. Luego, discute los hallazgos en grupos con otros compañeros y comparte sus ideas con el resto de la clase. Esta actividad promueve la habilidad para analizar críticamente el propio código, trabajar en equipo y desarrollar habilidades de revisión de código.

Evidencias:

Evidencia de Desempeño: AA1-EV02 – Revisión de código fuente:

- Documento con análisis crítico del código proporcionado, identificando mejoras y errores.
- Comentarios claros y descriptivos sobre las mejoras propuestas y los posibles problemas encontrados.
- Participación activa en la discusión grupal, compartiendo ideas y proporcionando retroalimentación constructiva a los compañeros.

Es de suma importancia utilizar metodologías de desarrollo en el proceso de construcción de software. Algunas de las metodologías más utilizadas incluyen Agile, Scrum, Waterfall y DevOps. Cada una de estas metodologías tiene sus propias ventajas.

Agile es conocida por su adaptabilidad y capacidad de respuesta a los cambios, así como por fomentar la colaboración y la participación activa del cliente. Scrum se destaca por su enfoque en la gestión de proyectos complejos a través de iteraciones y la autogestión del equipo. Waterfall ofrece un enfoque estructurado y bien definido, lo que facilita la planificación y el seguimiento del proyecto. DevOps se enfoca en la integración y colaboración continua entre los equipos de desarrollo y operaciones, acelerando la entrega y mejorando la calidad del software.

La elección de la metodología dependerá de las necesidades y características del proyecto, así como de las preferencias del equipo. Al utilizar metodologías de desarrollo adecuadas, se optimiza la planificación, se mejora la comunicación y se garantiza la entrega de software de alta calidad que cumple con las expectativas del cliente.

3.3.1.3 Construir una aplicación WEB: El aprendiz se desafía a construir una aplicación web de acuerdo al caso de estudio planteado para dar solución de software a una necesidad, utilizando el lenguaje de programación seleccionado y siguiendo las mejores prácticas de codificación. Utiliza una lista de requisitos y funcionalidades básicas proporcionadas, pero también fomenta su creatividad al agregar características adicionales. Durante el proceso de construcción, colabora con otros equipos y recibe retroalimentación periódica para discutir y mejorar el código. Al finalizar, presenta su aplicación web y comparte los desafíos y lecciones aprendidas durante el desarrollo. Esta actividad promueve la aplicación práctica de los conocimientos, el trabajo en equipo y la habilidad para presentar resultados de manera efectiva.

Evidencias:

Evidencia de producto: AA1-EV03 – Construcción de una aplicación WEB:

- Aplicación web funcional y bien estructurada, desarrollada siguiendo las mejores prácticas de codificación.
- Implementación exitosa de los requisitos y funcionalidades básicas, demostrando comprensión y habilidad para aplicar los conceptos aprendidos.
- Participación en las revisiones periódicas, mostrando mejoras realizadas en base a la retroalimentación recibida y explicando los desafíos enfrentados durante el proceso de desarrollo.

Duración: 236 Horas

4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento: **Evidencias de Desempeño:** Evidencia de Desempeño: AA1-EV02 – Revisión de código fuente	Analiza y comprende el código fuente suministrado, propone mejoras y corrige errores	Lista de chequeo
Evidencias de Producto: Evidencia de producto: AA1-EV01 – Planificación del proyecto	Planifica de forma correcta las actividades a desarrollar durante el ciclo de vida de desarrollo del proyecto	Lista de chequeo
Evidencia de producto: AA1-EV03 – Construcción de una aplicación WEB	Construye la aplicación web requerida siguiendo metodologías de desarrollo y buenas prácticas de diseño	Lista de chequeo

5. GLOSARIO DE TÉRMINOS

Desarrollo de software: Proceso de crear, diseñar, programar, probar y mantener software de aplicaciones o sistemas.

Lenguaje de programación: Un conjunto de instrucciones y reglas utilizadas para escribir programas informáticos.

Aplicación web: Un software que se ejecuta en un navegador web y que permite a los usuarios realizar diferentes tareas o acceder a servicios a través de Internet.

Mejores prácticas: Conjunto de técnicas, metodologías y enfoques reconocidos como los más efectivos y eficientes para lograr resultados óptimos en un determinado campo.

Metodología de desarrollo: Un enfoque estructurado y sistemático utilizado para gestionar y controlar el proceso de desarrollo de software.

Agile: Una metodología de desarrollo ágil que se enfoca en la adaptabilidad, la colaboración y la entrega incremental de software.

Scrum: Un marco de trabajo ágil que se basa en la colaboración y el trabajo en equipo, utilizando iteraciones llamadas "sprints" para gestionar proyectos complejos.

Waterfall: Una metodología de desarrollo tradicional y secuencial en la que el proceso se divide en fases claramente definidas y lineales.

DevOps: Una metodología que combina desarrollo y operaciones, buscando una integración y colaboración continua entre los equipos para acelerar la entrega de software.

Ciclo de vida del desarrollo de software: Las diferentes etapas y actividades que conforman el proceso de desarrollo de software, desde la concepción y planificación hasta el despliegue y mantenimiento.

Requisitos funcionales: Las funcionalidades y características específicas que debe tener un software para cumplir con los objetivos y necesidades del usuario.

Requisitos no funcionales: Los aspectos técnicos y de rendimiento que deben cumplirse en un software, como la seguridad, la escalabilidad y la usabilidad.

Codificación: El proceso de traducir los requisitos y diseños del software en código fuente legible por la máquina.

Pruebas de software: Actividades realizadas para verificar y validar que el software funciona correctamente y cumple con los requisitos establecidos.

Mantenimiento de software: Las actividades realizadas después de la implementación para corregir errores, mejorar el rendimiento y adaptar el software a los cambios y actualizaciones.

6. REFERENTES BILBIOGRÁFICOS

Minnick Jessica , (2021). Responsive Web design with HTML5 & CSS. Cengage Learning. https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=22936

Torres Remon, M. . (2018). Diseño web con HTML5 y CSS3. Macro. https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=6245

Rubiales Gómez, M. (2021). Curso de desarrollo Web. HTML, CSS y JavaScript. Edición 2021. Anaya Multimedia. https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=26733

Cabezas Granado, L. M. G. L. (2021). Curso de PHP 8 y MySQL 8. Anaya Multimedia. https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=26726

Escarcena, M. (2020). Programación páginas web JavaScript y PHP. Ra-Ma Editorial. https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=17194

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha

Autor (es)	Michel Osorio Barrera	Instructor	CCIT	15/04/2023

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	**Cargo**	**Dependencia**	**Fecha**	**Razón del Cambio**
Autor (es)					