



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



Universidad de las Fuerzas Armadas ESPE

Departamento de Ciencias de la Computación

Carrera: Ingeniería en Tecnologías de la
Información

Materia: Ingeniería de Software II

NRC 10523

Proyecto Extra

Alumno: Llumiquinga Llumiquinga Luis Miguel

Tutor: Ing. Efraín Fonseca Carrera

Fecha: 20 de agosto del 2023

Tabla de contenido

Ingeniería de software.....	3
Tema 6 3	
Caso practico 3	
6.1. Planteamiento de un caso de estudio	3
6.2 Desarrollo del caso de estudio	4
Ingeniería de Software basada en capas	5
login.py	6
sms.py.....	6
Evolución del Software	6
Verificación y validación del software	7

Tabla de Figuras

Figura 1. Funcionalidad del aplicativo. Fuente propia.....	3
Figura 2. Diagrama de secuencia del aplicativo. Fuente propia	4
Figura 3. Arquitectura de la aplicación. Fuente propia	5
Figura 4. Estructura de la aplicación. Fuente propia	5
Figura 5. Vista del aplicativo para las pruebas. Fuente propia	7

Ingeniería de software

Tema 6

Caso practico

6.1. Planteamiento de un caso de estudio

La Escuela de Fe y Alegría ubicada en el cantón Tachina provincia de Esmeraldas es una institución educativa que desempeña un papel fundamental en el acceso a la educación de calidad para niños y jóvenes de comunidades en situación de vulnerabilidad. Sin embargo, la gestión de la información de los estudiantes ha sido un desafío constante para la institución. Actualmente, el proceso de registro, seguimiento y acceso a los datos de los estudiantes se realiza de manera manual, lo que ha llevado a la pérdida de información, retrasos en la toma de decisiones y una ineficiente asignación de recursos. Para resolver estos problemas y mejorar la calidad de la educación brindada, se propone el desarrollo de un sistema de registro de estudiantes integral y seguro.

La funcionalidad del aplicativo esta expresada en la Figura 1.

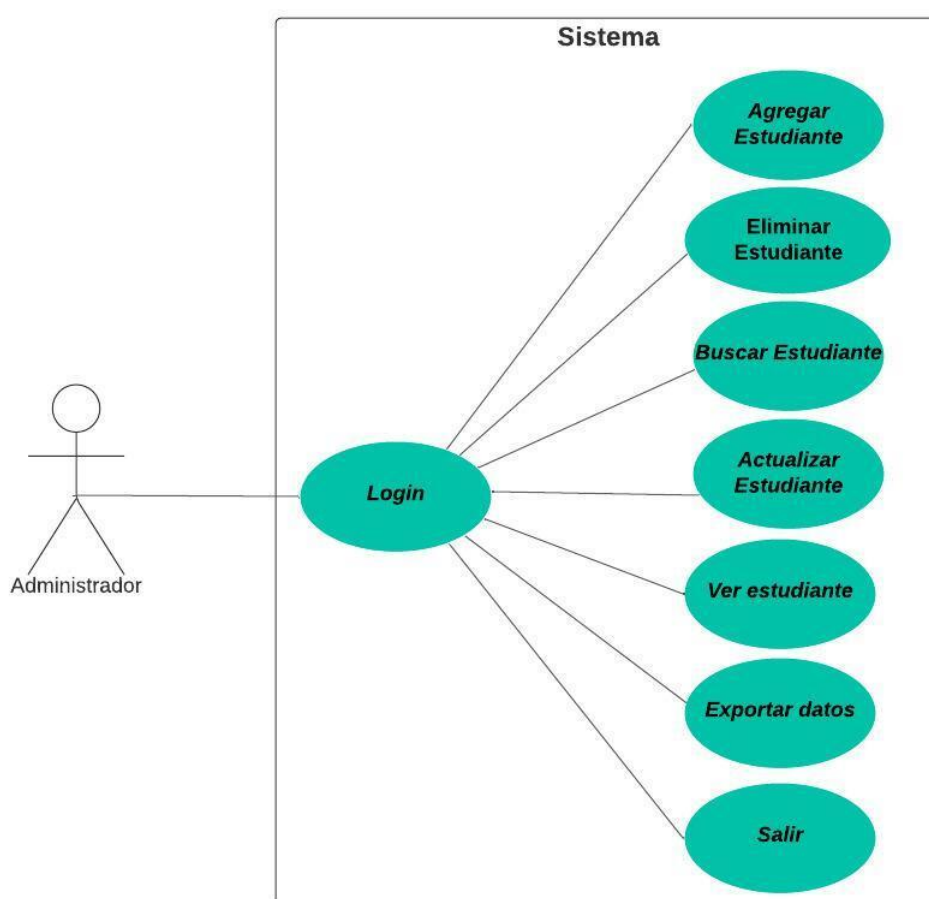


Figura 1. Funcionalidad del aplicativo. Fuente propia

Luego de que el administrador se autentifique con sus datos, seleccionara la operación que desee realizar, se ejecutan una secuencia de actividades (peticiones) de manera inmediata. Por ejemplo, si se trata de Agregar un Estudiante, el proceso es como sigue (ver Figura 2):

1. El administrador selecciona la acción a realizar
2. Se realiza un llamado a la función para realizar la acción
3. Se realiza el ingreso de los datos del alumno
4. Se retorna en pantalla los datos ingresados
5. El administrador observa los datos ingresados en una tabla

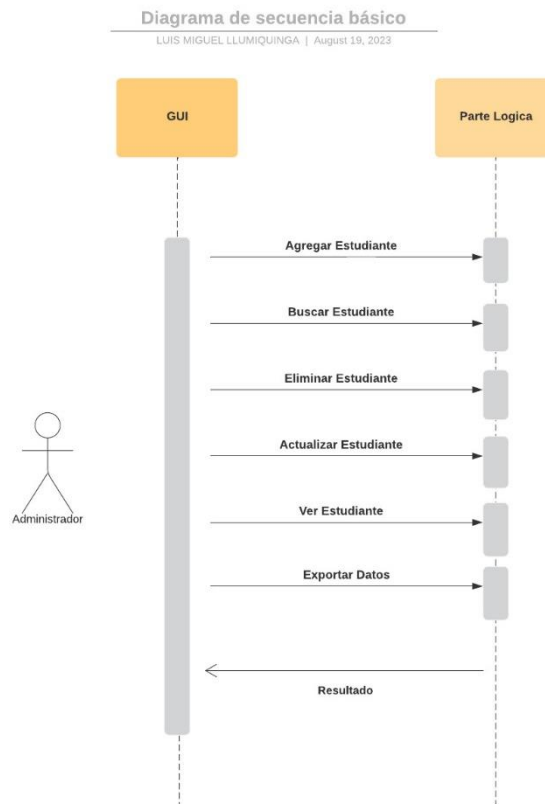


Figura 2. Diagrama de secuencia del aplicativo. Fuente propia

Con el propósito de poner en práctica el conocimiento adquirido en la unidad se plantea el caso de estudio descrito, cuyo producto es la implementación de una aplicación (y su mantenimiento) que permite realizar la gestión de alumnos haciendo uso del lenguaje de programación Python, IDE Pycharm y una base de datos MySQL.

6.2 Desarrollo del caso de estudio

El diseño de la aplicación para realizar la gestión de alumnos se encuentra acorde a la arquitectura mostrada en la Figura 3.

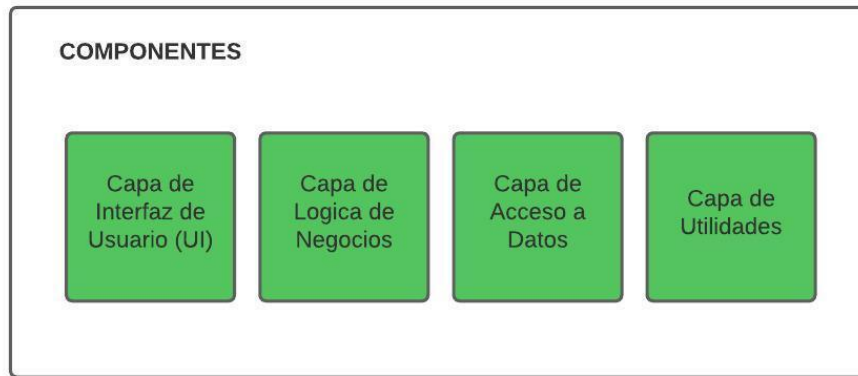


Figura 3. Arquitectura de la aplicación. Fuente propia

La arquitectura de la aplicación se organiza en capas que separan la interfaz de usuario, la lógica de negocios y el acceso a la base de datos. Esto permite una mejor organización del código y facilita la modificación y el mantenimiento en el futuro.

Ingeniería de Software basada en capas

La arquitectura de la aplicación se asemeja a una arquitectura de capas, donde las diferentes partes del código están organizadas en capas distintas, cada una con su propia responsabilidad. (ver Figura 4)

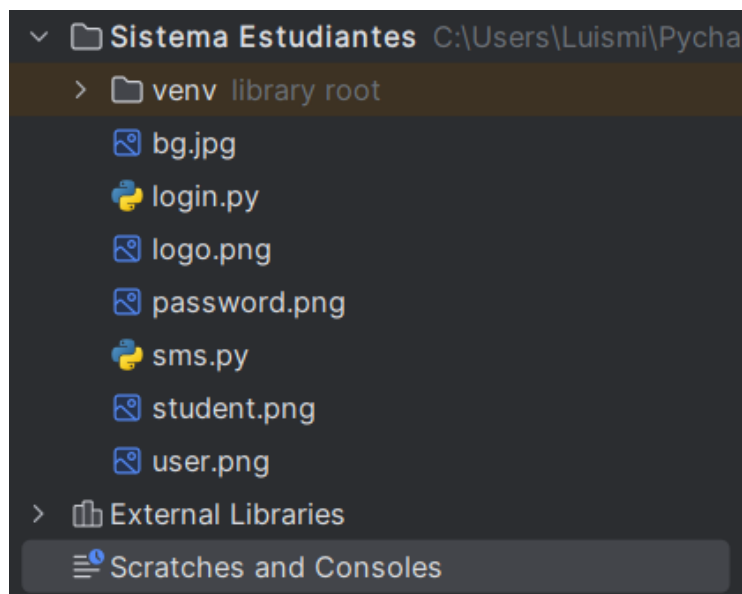


Figura 4. Estructura de la aplicación. Fuente propia

A continuación, se muestran las funciones y login relacionados, las cuales son implementadas en la aplicación.

login.py

```
login.py x sms.py
27 bgLabel.place(x=0, y=0)
28
29 loginFrame=Frame(window, bg='white')
30 loginFrame.place(x=400, y=150)
31
32 logoImage=PhotoImage(file='logo.png')
33
34 logoLabel=Label(loginFrame, image=logoImage)
35 logoLabel.grid(row=0, column=0, columnspan=2, pady=10)
36 usernameImage=PhotoImage(file='user.png')
37 usernameLabel=Label(loginFrame, image=usernameImage, text='Usuario', compound=LEFT, font=('times new roman', 20, 'bold'), bg='white')
38 usernameLabel.grid(row=1, column=0, pady=10, padx=20)
39
40 usernameEntry=Entry(loginFrame, font=('times new roman', 20, 'bold'), bd=5, fg='royalblue')
41 usernameEntry.grid(row=1, column=1, pady=10, padx=20)
42
43 passwordImage=PhotoImage(file='password.png')
44 passwordLabel=Label(loginFrame, image=passwordImage, text='Contraseña', compound=LEFT, font=('times new roman', 20, 'bold'), bg='white')
45 passwordLabel.grid(row=2, column=0, pady=10, padx=20)
46
47 passwordEntry=Entry(loginFrame, font=('times new roman', 20, 'bold'), bd=5, fg='royalblue')
48 passwordEntry.grid(row=2, column=1, pady=10, padx=20)
49
50 loginButton=Button(loginFrame, text='Login', font=('times new roman', 20, 'bold'), width=15, fg='white', bg='cornflowerblue', activebackground='white')
51 loginButton.grid(row=3, column=1, pady=10)
52 window.mainloop()
```

sms.py

```
login.py x sms.py x
1 from tkinter import *
2 import ttkthemes
3 from tkinter import ttk, messagebox, filedialog
4 import time
5 import pymysql
6 import pandas
7
8 #functional Part
9
10 usage
11 def iexit():
12     result=messagebox.askyesno('CONFIRMACION', 'Desea salir?')
13     if result:
14         root.destroy()
15     else:
16         pass
17 usage
18 def export_data():
19     url=filedialog.asksaveasfilename(defaultextension='.csv')
20     indexing=studenTable.get_children()
21     newList=[]
22     for index in indexing:
23         content=studenTable.item(index)
24         datalist=content['values']
25         newList.append(datalist)
26
27 table=pandas.DataFrame(newList, columns=['Id', 'Nombre', 'Teléfono', 'Email', 'Dirección', 'Género', 'Compleaños', 'Fecha Agregada'])
```

Evolución del Software

Es importante mencionar que el mantenimiento de los aplicativos de software es fundamental. Para este caso va desde la creación de una interfaz básica hasta la implementación de características adicionales como la gestión de base de datos de estudiantes, la interacción con una base de datos y elementos visuales como texto en movimiento y hora actualizada.

Verificación y validación del software

Esta etapa es importante ya que en este punto se verifica que la aplicación cumpla con su objetivo. En este caso, se realizan las pruebas con la finalidad de verificar que los procesos sean correctos.

Este proceso se lo puede realizar de manera visual una vez que la vista sea generada, como se puede observar en la Figura

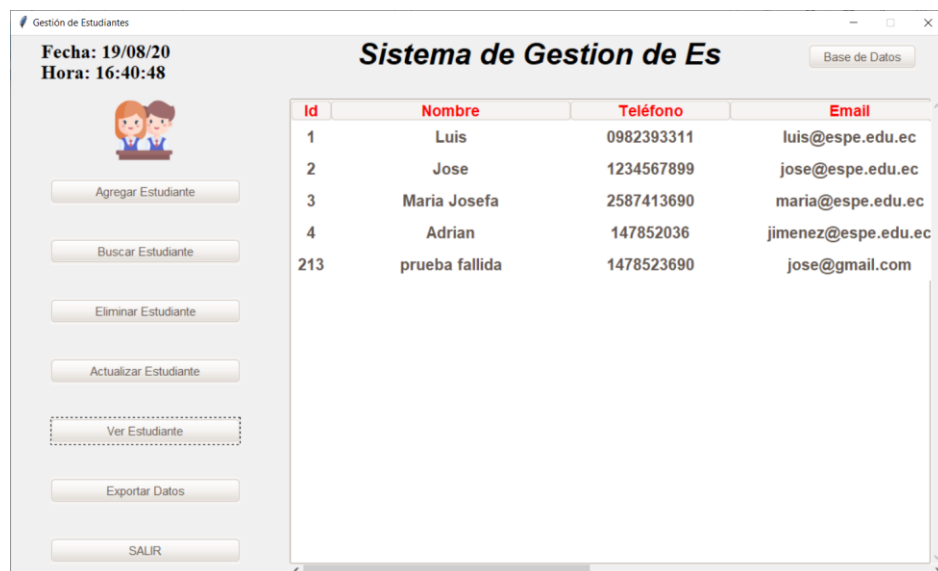


Figura 5. Vista del aplicativo para las pruebas. Fuente propia

A partir de lo expuesto se puede implementar nuevas funcionalidades o modificar estilos y vistas en función a la información recopilada, de tal manera que se garantice el correcto almacenamiento de los datos de los alumnos. Es decir, en base a este prototipo basico se puede modificar la interfaz (vista) y lógica con una justificación previa.