

# 3C: Company System With Face Recognition To Log

Isaac Barrios Campos, Brandon Reyes Alvarado Isriom@estudiantec.cr brandonra18@gmail.com

Área académica de Ingeniería en computadores

Instituto Tecnológico de Costa Rica

**Resumen**—Documentación del tercer proyecto de taller de programación. El proyecto consiste en la creación de un programa que administre los recibos, los trabajadores, los servicios ofrecidos y el reporte final. Además el proyecto debe contar con un acceso por medio de reconocimiento facial. Se ha utilizado tkinter para la creación de interfaces gráficas.

**Palabras clave**—Data manager, Proyecto, Taller de programación, Login Recognition.

## I. INTRODUCCIÓN

### I-A. Contexto

El sistema busca virtualizar todo los procesos que realizan los trabajadores de oficina. Esto incluye la creación de facturas, su consulta y el manejo de los servicios ofrecidos por la empresa. Además se agrega una opción que permite a un trabajador iniciar sección por medio de reconocimiento facial y la opción de que el equipo de TI registre nuevos usuarios en la aplicación para el reconocimiento facial Se busca aplicar conceptos de recursividad, iteración, almacenamiento en disco y clases.

### I-B. Problema

Analizar los conceptos vistos en clase y aplicarlos, encontrando soluciones en algoritmos, para el correcto funcionamiento del sistema. Desarrollar habilidades de investigación que permitan desarrollar y adaptarlo un tema al la necesidad del cliente.

### I-C. Objetivos y esbozo de solución

Diseñar una solución computacional relacionada con conceptos de recursividad, iteración, clases y almacenamiento en disco con el fin de cumplir con las necesidades de una empresa en el ámbito informático.

Para lograr esto se plantea crear una interfaz gráfica con la cual se obtengan los valores necesarios para la administración y que también permita a los trabajadores ingresar los datos específicos de cada factura despues de acceder por medio del reconocimiento facial. Un desglose se puede ver en fig.1.

## II. ANTECEDENTES O ESTADO DEL ARTE

Para este proyecto se Usaron las librerías de Tkinter, typing, fpdf, cv2, os, datetime, PIL y facerecognition.

Tkinter es una librería incluida dentro de python que permite graficar a partir de widgets, los cuales son el nombre que

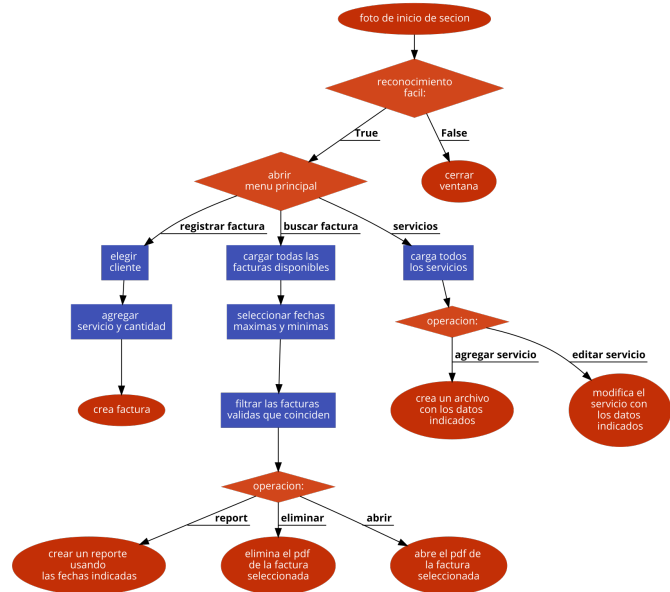


Figura 1. Boceto de solución

reciben los objetos en tkinter [8]. A esta librería se le ha agregado la extensión ttk que agrega mas widgets. [9]

typing es una librería incluida en python que se encarga de dar formato, de ella se ha usado el modulo Iterable que permite acceder a clases de datos para cada tipo de variable, esto permite estandarizar las entradas del sistema. [10]

fpdf es una librería para trabajar con archivos de formato pdf, con ella se crean los pdf. [4]

cv2(Open cv) es una librería para la utilización de cámaras y otros hardware de visión. [3]

os es una librería incluida dentro de python que se encarga del intercambio de información con el sistema operativo. [6]

datetime es una librería incluida en python para trabajar con el tiempo en formato de calendario gregoriano. [7]

PIL(pillow) es una librería incluida en python que permite trabajar con archivos de imágenes [2]

facerecognition es la librería a utilizar para el reconocimiento facial, esta es la interfaz entre python y el modelo de reconocimiento entrenado [1]

## III. PROPUESTA DE LA SOLUCIÓN

Para empezar con el proyecto se debe considerar la figura 1 que nos dice que ocuparemos el sistema de login, y el menú principal.

A su vez para el sistema de login se necesitara de dos partes, la obtención de una imagen por medio de la cámara y la parte que compara la imagen obtenida con las imágenes en la base de datos.

Para la obtención de la imagen se puede hacer uso de CV2 [3] que nos permitirá guardar un frame de la cámara. Este captura sera temporal y deberá tener un nombre específico para diferenciarle de las imágenes de los usuarios registrados.

Una vez obtenida la imagen procederemos a cargar los datos de los usuarios registrado y a crear su encoding facial. Para ello obtenemos el nombre de todas las imágenes en la base de datos (excluyendo la imagen tomada) con os [6] y las procesamos con el encoding de face recognition [1]. Obtenidos ambos los almacenamos en memoria.

Después de esto obtenemos el face encoding de la imagen tomada la comparáramos con las lista que almacena los encodings de todos los usuarios registrados. El método para realizar esto en face recognition [1] devuelve una lista donde indica la coincidencia con usuario, de ahí podemos deducir si esta iniciando seccion un usuario registrado y su nombre, para mejor entendimiento ver figura 2.

```
def login():
    """
    Check all the images in the database and compare with the img form login image
    :return: a tuple that indicate if user is in the database and user
    """
    # -----
    # This sentences will load and encode the image and face of the person trying to login
    image_of_user = face_recognition.load_image_file(
        "user_database/user_0.png")
    user_face_encoding = face_recognition.face_encodings(image_of_user)[0]
    # -----
    # This compare the face encoding with all users encoding's
    results = False
    if True in face_recognition.compare_faces(faces, user_face_encoding, 0.469):
        results = True
    # -----
    # Print the results
    if results:
        print("Welcome!")
        return True, face_recognition.compare_faces(faces, user_face_encoding, 0.469).index(True)
    else:
        print("You are not in our database! Please create your profile or get in contact with support")
        return False, -1
```

Figura 2. Función que compara los rostros

En caso de que el encoding no coincida con ninguno de los que están almacenados en la base de datos, automáticamente enviará al usuario a una ventana de registro, donde podrá introducir sus datos e imagen para registrarse en la base de datos. Véase en figura 3

Figura 3. Menú de registro

Una vez creado el sistema de login se procede a crear el menú principal. Para la parte gráfica del sistema se usaran

ventanas de tkinter [8] y clases de python, ya que sera posible tener varias ventanas abiertas y todas tendrán elemento similares:

- Titulo con el nombre de la empresa
- Logo de la empresa en la parte superior
- Tamaño fijo para no obstaculizar toda la pantalla

En el caso específico del menú principal se usara una plantilla como la de la figura 4.

Figura 4. plantilla del menú principal

### III-A. Crear factura

La creación de facturas permitirá crear un .txt permanente en la base de datos y un .pdf para el cliente que servirá para reconocer si la factura esta en débito. Para la creación de la interfaz de esta sección se uso una plantilla similar a la de la figura 5. En esta se podrán ingresar los datos del cliente, mientras que los datos de la factura (numero, fecha, y fecha limite) serán generados automáticamente. Además para el ingreso de datos del servicio se provee de dos campos, el de service que cuenta con selección y el de quantity que permitirá ingresar la cantidad, los campos de precio del servicio y sub total se rellenaran automáticamente conforme a los otros dos campos.

Junto a estos se encuentran dos botones que permitirán agregar la configuración de actual servicios al listado de la factura o eliminar la información de estos o del servicio listado seleccionado, en caso de que alguno este seleccionado. Además esta habilitado la opción de traer la información desde el listado de servicios de la factura hasta los espacios de servicio para duplicar registro fácilmente. Todos estos espacio son widgets de tkinter con eventos [5] o de ttk [9] como puede ser el caso del treeview o las combobox. A esta plantilla es necesario agregar dos botones para terminar la factura o para cancelarla. El primer botón también creara tanto el pdf como el .txt en base a el listado de servicios.

A demás para la creación y modificación de servicios se puede realizar algo similar, ingresando los datos del servicio en vez de los de factura y de cliente.

### III-B. Buscar recibos

Para la ventana de búsqueda de recibos se ha utilizado también una plantilla, similar a la figura 6. Esta ventana permitirá

## 3C Company

Customer data:

Customer name:

Customer email:

Bill N°:

Date:

Due date:

Service	Quantity	Price	Sub Total

Service	Quantity	Sub total	Amount

Figura 5. plantilla del Crear Factura

buscar facturas por fechas, eliminar facturas, y crear reportes según un rango de fechas. Para ello la ventana cuenta con una sección en donde se seleccionaran las fechas y conforme esta información se visualizaran las facturas pertinentes. Para la obtención de los datos que deberán estar en el listado se hace uso de los archivos .txt creados en la sección III-A usando como segundo filtro de selección el que exista su contra parte como pdf ya que esto indicara que la factura aun esta pendiente, para la creación del reporte este segundo filtro no existirá. Además se ha de calcular el total de estas facturas y crear los botones para realizar cada paso.

[illegible]

Figura 6. plantilla del búsqueda de factura

## IV. RESULTADOS

Las ventanas de las secciones III-B y III-A se pueden ver en las figuras 8 y 9 respectivamente. Además se puede ver el diagrama de clases del proyecto en la figura 7

## V. CONCLUSIONES

En el proyecto se lograron aplicar los conocimientos vistos en clase para implementar una solución profesional al problema, alcanzando un resultado útil y aplicable en un contexto laboral, igualmente se logran alcanzar los objetivos propuestos desde el inicio del proyecto.

También nos deja objetivos por aspirar, como implementar GUIs más fluidas y optimizadas, con mejores diseños, al igual

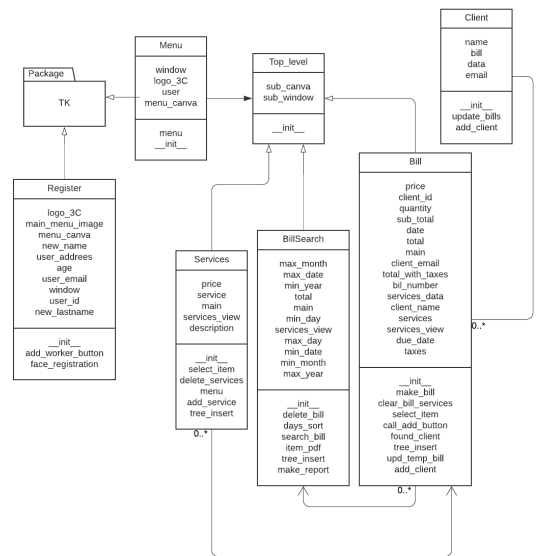



Figura 7. Diagrama de clases



THE 3C COMPANY

CREATE | CARE | CONSERVE

Customer data:

Customer name:

Customer email:

Bill N°: 2

Date: 2020-08-12

Due date: 2020-08-14

Service	Quantity	Price	Sub Total	X
<input type="text" value="Alejandro"/>	1	128	128	1


  

Service	Quantity	Sub total	Amount
Alejandro	152	19456	20992
Alejandro	11	1408	taxes
Alejandro	1	128	2729.96

Total

23720

Figura 8. ventana de búsqueda



THE 3C COMPANY

CREATE | CARE | CONSERVE

MIN

Dates

MAX

Figura 9. Ventana de facturación

el diseño de los informes y PDFs. Dentro del código se crean funciones que difícilmente se llegan a usar, por lo tanto hallar una manera de involucrar estas entre las demás funciones es un objetivo de optimización a mejorar.

## REFERENCIAS

- [1] A. Geitgey, "Face Recognition — Face Recognition 1.2.3 documentation", Face-recognition.readthedocs.io, 2020. [Online]. Available: <https://face-recognition.readthedocs.io/en/latest/readme.html>. [Accessed: 13- Aug- 2020].
- [2] F. Lundh and A. Clark, "Pillow — Pillow (PIL Fork) 7.2.0 documentation", Pillow.readthedocs.io, 2020. [Online]. Available: <https://pillow.readthedocs.io/en/stable/>. [Accessed: 13- Aug- 2020].
- [3] Open Source Computer Vision, "OpenCV: OpenCV modules", Docs.opencv.org, 2020. [Online]. Available: <https://docs.opencv.org/master/index.html>. [Accessed: 13- Aug- 2020].
- [4] M. Pat, "PyFPDF", Pyfpdf.readthedocs.io, 2020. [Online]. Available: <https://pyfpdf.readthedocs.io/en/latest/>. [Accessed: 13- Aug- 2020].
- [5] F. Lundh, "An Introduction to Tkinter (Work in Progress)", Effbot.org, 2014. [Online]. Available: <https://effbot.org/tkinterbook/>. [Accessed: 13- Aug- 2020].
- [6] Python Software Foundation, "os — Miscellaneous operating system interfaces — Python 3.8.5 documentation", Docs.python.org, 2020. [Online]. Available: <https://docs.python.org/3/library/os.html>. [Accessed: 13- Aug- 2020].
- [7] Python Software Foundation, "datetime — Basic date and time types — Python 3.8.5 documentation", Docs.python.org, 2020. [Online]. Available: <https://docs.python.org/3/library/datetime.html>. [Accessed: 13- Aug- 2020].
- [8] Python Software Foundation, "Graphical User Interfaces with Tk — Python 3.8.4 documentation", Docs.python.org, 2020. [Online]. Available: <https://docs.python.org/3/library/tk.html>. [Accessed: 13- Aug- 2020].
- [9] Python Software Foundation, "tkinter.ttk — Tk themed widgets — Python 3.8.5 documentation", Docs.python.org, 2020. [Online]. Available: <https://docs.python.org/3/library/tkinter.ttk.html>. [Accessed: 13- Aug- 2020].
- [10] Python Software Foundation, "typing — Support for type hints — Python 3.8.5 documentation", Docs.python.org, 2020. [Online]. Available: <https://docs.python.org/3/library/typing.html>. [Accessed: 13- Aug- 2020].