

Sistema de gestión de horas laboradas y detección facial con Python y Flask

Franklin Daniel Casallas Vanegas
Kelly Johanna Castañeda García

Ingeniería de Sistemas - Modalidad Virtual, Fundación universitaria del Área Andina.

*Diplomado Programación Python y DevNet
Bogotá, Colombia.*

fcasallas@estudiantes.areandina.edu.co
kcastaneda8@estudiantes.areandina.edu.co

Resumen

Este trabajo presenta el resultado de un sistema que se desarrolla para la gestión de informes tanto para los trabajadores como para los administradores del sistema, esto suplente la necesidad de estar solicitando informes de tiempo laborado a las áreas de gestión humana y poder realizar seguimiento de los trabajadores sobre sus tiempos en las empresas. También se implementa un sistema de detección facial para que en la ampliación del proyecto se pueda realizar el entrenamiento y cumpla con el recogimiento facial de los usuarios.

El propósito del sistema, es la gestión automatizada de reportes de días laborados, tiempo que se emplea al día, con el fin de realizar el acompañamiento pertinente a cada uno de los trabajadores en el cual se implemente el sistema.

Palabras clave

1. **Control:** Es la acción que se realiza sobre el sistema con el fin de ir corrigiendo inconvenientes.
2. **Registro:** Apuntes necesarios para realizar correctamente el seguimiento al proyecto, tiene como finalidad la continua mejora.
3. **Seguimiento:** Acción que se realiza desde el inicio del proyecto, con el fin de poder entregar el producto funcional.
4. **Python:** Lenguaje de programación que se utilizó para realizar toda la lógica del sistema, se complementa con las diferentes librerías según su funcionalidad.
5. **Sistema de información:** Desarrollo que se implementa para suplir una necesidad.

Abstract

This work presents the result of a system that is developed for the management of reports for both workers and system administrators, this supplements the need to be requesting reports of time worked to the areas of human management and to track workers on their times in the companies. A facial detection system is also implemented so that in the extension of the project the training can be carried out and comply with the users' facial collection.

The purpose of the system is the automated management of reports of days worked, time spent per day, in order to perform the relevant monitoring of each of the workers in which the system is implemented.

Keywords

1. **Control:** It is the action that is performed on the system in order to correct inconveniences.
2. **Registration:** Necessary notes to correctly follow up the project, with the purpose of continuous improvement.
3. **Follow-up:** Action that is carried out from the beginning of the project, in order to deliver the functional product.
4. **Python:** Programming language that was used to perform all the logic of the system, it is complemented with the different libraries according to their functionality. That is placed in the top box of the browsers.
5. **Sistema de información:** Desarrollo que se implementa para suplir una necesidad.

I. INTRODUCCIÓN

En la actualidad los sistemas de información han logrado facilitar de una u otra forma las actividades de administradores y usuarios; aun cuando ya existen múltiples plataformas que almacenan y son de fácil uso, sin embargo día a día sigue surgiendo la necesidad de buscar alternativas para agilizar procesos en los que se intenta automatizar cada vez más estas tareas administrativas y de seguimiento, de esta necesidad diaria es que surge la iniciativa de crear un sistema que no solo fuera práctico, sino que también sea de fácil uso y al mismo tiempo eficiente, buscando siempre satisfacer al cliente. Este sistema de gestión tiene como objetivo apoyar y facilitar el seguimiento, tanto en el registro de los horarios como en su seguimiento y consulta, con el fin de garantizar siempre la facilidad de uso.

II. METODOLOGÍA

Creación de ambiente de trabajo en Windows.

En esta sección se hace una breve descripción de las condiciones del ambiente ideal que debe tener el usuario para el uso del sistema.

Instalación de Python

Se especifica que la instalación que se propone es en ambiente Windows.

Se descarga de la página oficial de Python el ejecutable y se instala en el equipo. Para el caso de este proyecto acepta Python 3 en cualquiera de sus versiones.

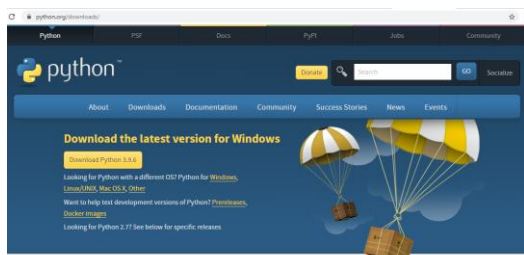


Ilustración 1. Sitio Oficial de Python, Descargas. Recuperada de: <https://www.python.org/downloads/>

Para esta actividad se utilizó Python 3.8.

Instalación visual studio code

Se realiza instalación para el desarrollo de la versión dado que Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para depuración, control Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos y refactorización de código

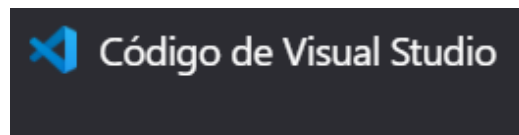


Ilustración 2. Creación del entorno Virtual. Fuente propia.

Una vez instalado se procede a empezar el desarrollo, se crea infraestructura de carpetas que tendrá el sistema.

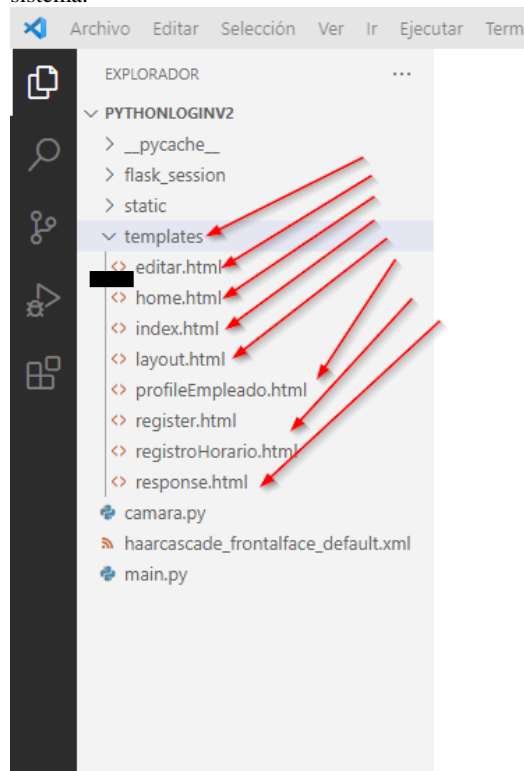


Ilustración 3. Creación del entorno Virtual. Fuente propia.

Instalación de Librerías

Ya estando dentro del entorno virtual se deben instalar las librerías necesarias tanto para el desarrollo como para la ejecución del sistema.

Pip

Gestor de paquetes, que se usa para la instalación y administración de software en Python. En caso de no tenerlo instalado, se debe descargar e instalar.

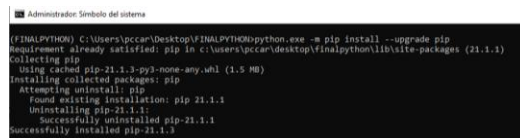


Ilustración 4. Actualización de pip. Fuente propia.

Librería bs4

Esta es una librería de Python para extraer datos html.

```

Administrador: Símbolo del sistema

(FINALPYTHON) C:\Users\pccar\Desktop\FINALPYTHON>pip install bs4
Collecting bs4
  Using cached bs4-0.0.1.tar.gz (1.1 kB)
Collecting beautifulsoup4
  Using cached beautifulsoup4-4.9.3-py3-none-any.whl (115 kB)
Collecting soupsieve>1.2
  Using cached soupsieve-2.2.1-py3-none-any.whl (33 kB)
Using legacy 'setup.py install' for bs4, since package 'wheel' is not installed.
Installing collected packages: soupsieve, beautifulsoup4, bs4
Running setup.py install for bs4 ... done
Successfully installed beautifulsoup4-4.9.3 bs4-0.0.1 soupsieve-2.2.1

```

Ilustración 5. Instalación de la librería bs4. Fuente propia.

Librería flask

Flask es un “micro” Framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC.

Flask-Jsonpify	1.5.0
Flask-Login	0.5.0
Flask-MySQL	1.5.2
Flask-MySQLdb	0.2.0
Flask-Session	0.4.0

Ilustración 6. Instalación de la librería flask. Fuente propia.

Librería requests

Librería de Python para hacer peticiones http.

```

Administrador: Símbolo del sistema

(FINALPYTHON) C:\Users\pccar\Desktop\FINALPYTHON>pip install requests
Collecting requests
  Using cached requests-2.26.0-py3-none-any.whl (62 kB)
Collecting certifi~2017.4.17
  Using cached certifi-2017.4.17-py3-none-any.whl (145 kB)
Collecting idna~2.3
  Using cached idna-3.2-py3-none-any.whl (59 kB)
Collecting charset-normalizer~2.0.0
  Using cached charset-normalizer-2.0.0-py3-none-any.whl (35 kB)
Collecting urllib3~1.26.0
  Using cached urllib3-1.26.0-py3-none-any.whl (138 kB)
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
Successfully installed certifi-2017.4.17 idna-3.2 requests-2.26.0 urllib3-1.26.0

```

Ilustración 7. Instalación de la librería requests. Fuente propia.

Librerías usadas para cosas puntuales como abrir cámara y exportar datos en Excel.

PyYAML	5.4.1
requests	2.26.0
requests-oauthlib	1.3.0
rsa	4.7.2
scipy	1.7.0
selenium	3.141.0
setuptools	56.0.0
simplejson	3.17.3
six	1.15.0
soupsieve	2.2.1
tensorboard	2.5.0
tensorboard-data-server	0.6.1
tensorboard-plugin-wit	1.8.0
tensorflow	2.5.0
tensorflow-estimator	2.5.0
termcolor	1.1.0
timedelta	2020.12.3
tk	0.1.0
typing-extensions	3.7.4.3
typish	1.9.2
urllib3	1.26.6
virtualenv	20.5.0
Werkzeug	2.0.1
wheel	0.36.2
wrapt	1.12.1
xlrd	2.0.1
xlwt	1.3.0
xmltodict	0.12.0

Ilustración 8. Instalación de la librerías. Fuente propia.

Diagrama de bloques

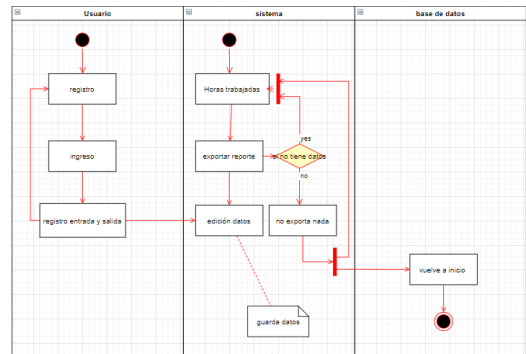


Ilustración 8. Diagrama General. Fuente propia.

Manual de usuario

Al ingresar al enlace encontraremos lo siguiente:

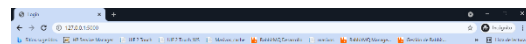


Ilustración 9. sistema de gestión horas. Fuente propia

Si aún no, no hemos registrado, procedemos hacerlo.

Registro

Ilustración 10. sistema de gestión horas. Fuente propia

Ilustración 11. sistema de gestión horas. Fuente propia

Ilustración 12. sistema de gestión horas. Fuente propia

Una vez registrado, ya puede proceder a ingresar al sistema

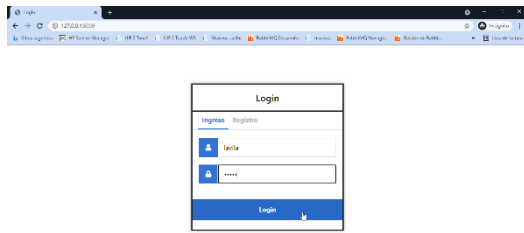


Ilustración 13. sistema de gestión horas. Fuente propia
Lo primero que muestra el sistema es la tabla de horas trabajadas, un filtro y el menú de navegación.

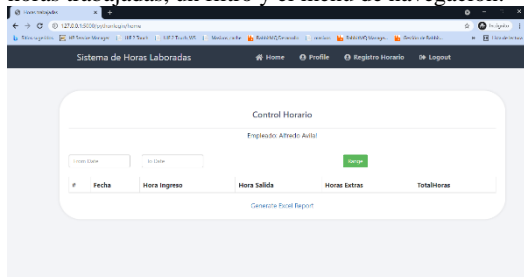


Ilustración 14. sistema de gestión horas. Fuente propia
Perfil

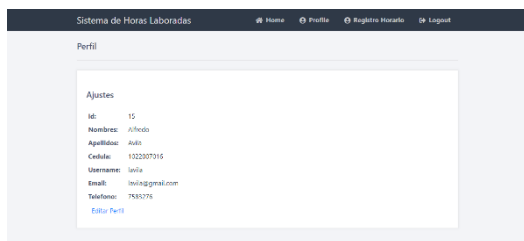


Ilustración 15. sistema de gestión horas. Fuente propia

Editar

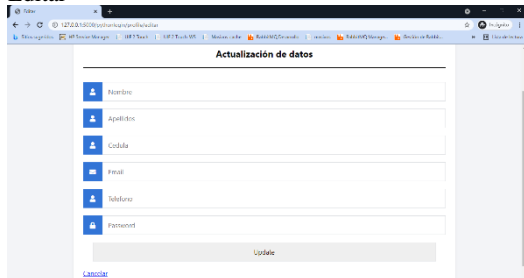


Ilustración 16. sistema de gestión horas. Fuente propia
Registro Horas

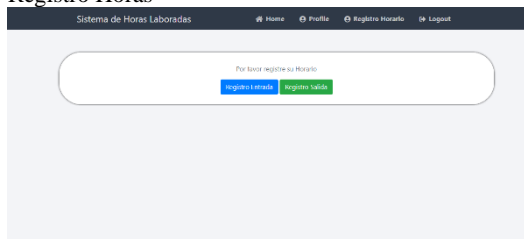


Ilustración 17. sistema de gestión horas. Fuente propia
Registro entrada

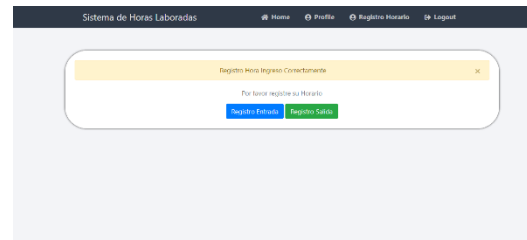


Ilustración 18. sistema de gestión horas. Fuente propia
Se valida en la tabla

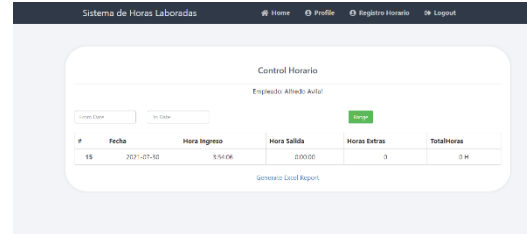


Ilustración 19. sistema de gestión horas. Fuente propia

Exportar

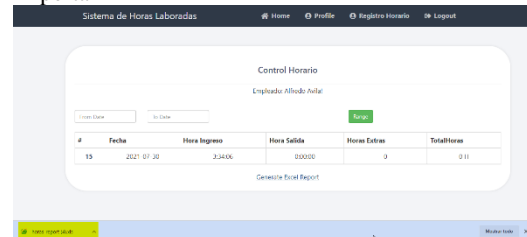


Ilustración 20. sistema de gestión horas. Fuente propia

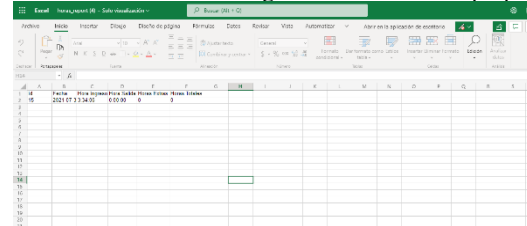


Ilustración 21. sistema de gestión horas. Fuente propia

Registrar salida

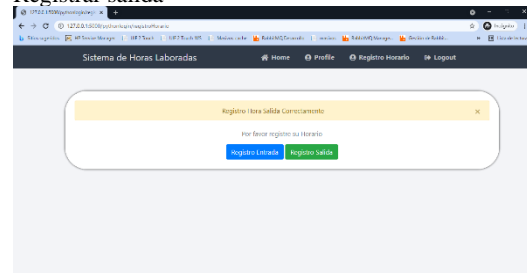


Ilustración 22. sistema de gestión horas. Fuente propia

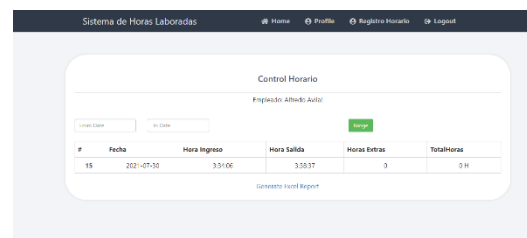


Ilustración 23. sistema de gestión horas. Fuente propia

[illegible]

En la clase lo primero que se hace es el llamado a todas las librerías necesarias y la conexión a base de datos.

```
File Edit Selección View Terminal Ayuda • main.py - python(708) - Visual Studio Code [Administrador]
PYTHONENVIRONM...
main.py
1 # -*- coding: utf-8 -*-
2
3 import time
4 import re
5 from datetime import datetime, timedelta
6 import cv2
7 from camera import Video
8 from flask_socketio import FlaskSocketIO
9 import numpy as np
10 from matplotlib import pyplot as plt
11 from sklearn.metrics import f1_score
12 from camera import Video
13 import sys
14 import io
15 import shutil
16 import pygame
17
18 ##Initia
19 mysql = MySQL()
20 app = Flask(__name__)
21 app.secret_key = "your secret key"
22
23 #Se configura la Base de datos
24 app.config['MYSQL_HOST'] = '192.168.1.10'
25 app.config['MYSQL_USER'] = 'root@localhost'
26 app.config['MYSQL_PASSWORD'] = ''
27 app.config['MYSQL_DB'] = 'seguridad_python_db'
28
29 #Se crea el objeto MySQL
30 mysql.init_app(app)
```

Por lo cual lo primero que se realiza es la instalación de los prerrequisitos y validamos que estén las librerías

protobuf	3.17.3
pyasn1	0.4.8
pyasn1-modules	0.2.8
pycparser	2.20
PyMySQL	1.0.2
pyparsing	2.4.7
python-csv	0.0.13
python-dateutil	2.8.2
pytz	2021.1
PyYAML	5.4.1
requests	2.26.0
requests-oauthlib	1.3.0
rsa	4.7.2
scipy	1.7.0
selenium	3.141.0
setuptools	56.0.0
simplejson	3.17.3
six	1.15.0
soupsieve	2.2.1
tensorboard	2.5.0
tensorboard-data-server	0.6.1
tensorboard-plugin-wit	1.8.0
tensorflow	2.5.0
tensorflow-estimator	2.5.0
termcolor	1.1.0
timedelta	2020.12.3
tk	0.1.0
typing-extensions	3.7.4.3
typish	1.9.2
urllib3	1.26.6
virtualenv	20.5.0
Werkzeug	2.0.1
wheel	0.36.2

Una vez realizado esto, se crea un método para cada opción del sistema con su respectivo .html.

[illegible]

Una vez creado los métodos y los archivos HTML, se procede a ejecutar aplicación y validar que funcione correctamente.

```
a GPU set up on your machine.
* Debugger is active!
* Debugger PIN: 125-640-729
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

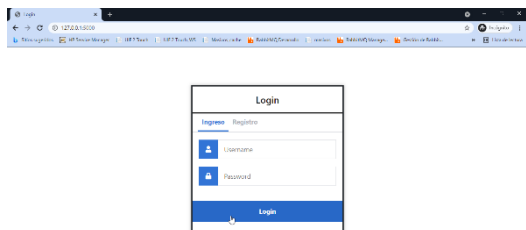


Ilustración 29. sistema de gestión horas. Fuente propia

III. RESULTADOS

Con el uso de métodos y las herramientas mencionadas anteriormente, pudimos obtener resultado con el sistema de registro de empleados y su manejo de datos.

IV. DISCUSIÓN

Los resultados obtenidos en este proyecto demuestran que la asociada esta lista para avanzar tecnológicamente hablando, hoy en día las compañías grandes y pequeñas buscan ese valor agregado que proporciona un sistema, siempre intentando agilizar e incluso automatizar sus procesos, todo pensando por un bien a largo plazo. Al realizar este sistema e investigando un poco es increíble de creer que todos los códigos utilizados se conocen hoy, son el resultado de la Segunda Guerra Mundial en la década de 1940. En esos años, el concepto de "sistema operativo" ni siquiera existía, y los programadores interactuaban directamente con el hardware de la computadora. Idioma (es decir, en binario, solo se utilizan 0 y 1 para la programación).

El concepto de sistema operativo apareció en la década de 1950. El primer sistema operativo de la historia se creó para la computadora IBM 704 en 1956. Básicamente, lo que hizo fue comenzar a ejecutar el programa cuando se completó el programa anterior.

En la década de 1960, se produjo una revolución en el campo de los sistemas operativos. Han aparecido los conceptos de sistema multitarea, sistema multiusuario, sistema multiprocesador y sistema en tiempo real.

Fue durante esta década que apareció UNIX, y es la base de la mayoría de los sistemas operativos que existen en la actualidad.

Gracias a estos pasos que en su momento fueron escandalosos es que hoy en día contamos con el privilegio de avanzar y conocer cada vez más sobre los sistemas operativos que existen y los que se pueden hacer.

Y pensar que cada sistema se genera pensando en una carencia, en nuestro caso particular, nos parecía algo útil, afocarnos en la creación de un sistema que ayudara no solo al administrador si no al mismo trabajador a llevar un control de su horario, más detallado y un poco más automático, se desarrollo este sistema pensando siempre en

ayudar al cliente, porque es importante mejorar en el control del horario laboral, y con este sistema se esta dando un paso a que esto pase.

V. CONCLUSIONES

Es posible que muchas empresas se pregunten cómo controlar las horas de trabajo para optimizar al máximo el rendimiento de sus empleados sin tener por ello que destinar más recursos de los estrictamente necesarios para esta función. Dado que ellos tienen claro que ningún sistema de cómputo es para siempre dado que con el tiempo requieran varias actualizaciones, hasta el punto de quedarse obsoleto, por esto es importante generar un buen proyecto con su documentación correcta, de esta manera se podrá garantizar que se gastaran menos recursos, dando la certeza que el proyecto fue basado completamente en cubrir las necesidades que lo originaron y lo mas importante que sea capaz de satisfacerlas.

Así que se puede concluir que El desarrollo de un sistema personalizado siempre es beneficioso para cualquier usuario, porque eso es lo que busca la empresa, a menos ajustes y mayor duración, más rentable es para él, porque siempre busca minimizar costos. Gastar dinero en su propio sistema.

VI. REFERENCIAS BIBLIOGRÁFICAS

- (Welcome to Python.org, 2021)
- [2] Kaspersky. (2020). Exploits y vulnerabilidades [Sitio oficial]. Recuperado de: <https://latam.kaspersky.com/resource-center/threats/malware-system-vulnerability>
- Mark Otto, a., 2021. Introduction. [online] Getbootstrap.com. Available at: <https://getbootstrap.com/docs/4.1/getting-started/introduction/> [Accessed 30 July 2021].
- Dev.mysql.com. 2021. MySQL :: MySQL Documentation. [online] Available at: <https://dev.mysql.com/doc/> [Accessed 30 July 2021].
- Flask.palletsprojects.com. 2021. Welcome to Flask — Flask Documentation (2.0.x). [online] Available at: <https://flask.palletsprojects.com/en/2.0.x/> [Accessed 30 July 2021].
- Jinja.palletsprojects.com. 2021. Jinja — Jinja Documentation (3.0.x). [online] Available at: <https://jinja.palletsprojects.com/en/3.0.x/> [Accessed 30 July 2021].
- Developer.mozilla.org. 2021. AJAX - Guía de Desarrollo Web | MDN. [online] Available at: <https://developer.mozilla.org/es/docs/Web/Guide/AJAX> [Accessed 30 July 2021].
- Fib.upc.edu. 2021. Historia de los sistemas operativos. [online] Available at: <https://www.fib.upc.edu/retro-informatica/historia/%20Operativo%20de,campo%20de%20los%20Sistemas%20Operativos.> [Accessed 30 July 2021].