

**HERRAMIENTAS EMERGENTES
PARA LA CONSTRUCCIÓN DE UNA
APLICACIÓN WEB PARA LA
PRESTACIÓN DE SERVICIOS⁹²
EMERGING TOOLS FOR THE
CONSTRUCTION OF A WEB
APPLICATION FOR THE PROVISION
OF SERVICES**

Martínez Moreno Martha⁹³

Álvarez Vilchis Jonathan⁹⁴

Estrada Manuel Luis Antonio⁹⁵

Abundez Barrera Itzel María⁹⁶

Flores Olivares Luis Roberto⁹⁷

Pares evaluadores: Red de Investigación en Educación, Empresa y Sociedad – REDIEES.⁹⁸

⁹² Derivado del proyecto de investigación: RYM Aplicación web para servicios

⁹³ Profesor adscrito al departamento de sistemas y computación, Tecnológico Nacional de México, Instituto Tecnológico de Toluca. México.

⁹⁴ Estudiante de la carrera de Ingeniería en sistemas computacionales. Tecnológico Nacional de México, Instituto Tecnológico de Toluca. México.

⁹⁵ Profesor adscrito al departamento de sistemas y computación, Tecnológico Nacional de México, Instituto Tecnológico de Toluca. México.

⁹⁶ Profesor adscrito al departamento de sistemas y computación, Tecnológico Nacional de México, Instituto Tecnológico de Toluca. México.

⁹⁷ Profesor adscrito al departamento de sistemas y computación, Tecnológico Nacional de México, Instituto Tecnológico de Toluca. México.

⁹⁸ Red de Investigación en Educación, Empresa y Sociedad – REDIEES. www.rediees.org

8. HERRAMIENTAS EMERGENTES PARA LA CONSTRUCCIÓN DE UNA APLICACIÓN WEB PARA LA PRESTACIÓN DE SERVICIOS⁹⁹

Martínez Moreno Martha¹⁰⁰, Álvarez Vilchis Jonathan¹⁰¹, Estrada Manuel Luis Antonio¹⁰²,
Abundez Barrera Itzel María¹⁰³, Flores Olivares Luis Roberto¹⁰⁴

RESUMEN

En el Estado de México, así como en toda la república mexicana, las empresas y negocios dedicados a la prestación de servicios de construcción han tenido problemas para mantenerse en pie en estos últimos años. Debido a la disminución de oportunidades de trabajo y por los recortes del personal, desde las actividades administrativas hasta quienes realizan los servicios (albañiles, arquitectos, etc.), de la misma forma, los solicitantes de servicios de construcción, reparación y mantenimiento de edificios, al momento de buscar quien realice estos servicios, tienen probabilidades de ser víctimas de estafas. Y ambas partes son propensas a la informalidad en la entrega y del pago por el servicio.

Para un estudiante de la carrera de Ingeniería en sistemas computacionales, el desarrollo de habilidades y competencias tecnológicas implica el uso de tecnologías ágiles para la construcción de aplicaciones que permitan resolver un problema real. De esta forma, se crea una aplicación web y es el presente documento el que describe a RyM como una aplicación web creada con herramientas de desarrollo tecnológico con un auge reciente, cuyo objetivo es reactivar una zona económica y publicar los servicios de prestadores. En la cual los solicitantes podrán contactar a proveedores de servicios de confianza y cuenten con una previa evaluación de sus trabajos realizados.

⁹⁹ Derivado del proyecto de investigación: RYM Aplicación web para servicios

¹⁰⁰ Profesor adscrito al departamento de sistemas y computación, Tecnológico Nacional de México, Instituto Tecnológico de Toluca. México.

¹⁰¹ Estudiante de la carrera de Ingeniería en sistemas computacionales. Tecnológico Nacional de México, Instituto Tecnológico de Toluca. México.

¹⁰² Profesor adscrito al departamento de sistemas y computación, Tecnológico Nacional de México, Instituto Tecnológico de Toluca. México.

¹⁰³ Profesor adscrito al departamento de sistemas y computación, Tecnológico Nacional de México, Instituto Tecnológico de Toluca. México.

¹⁰⁴ Profesor adscrito al departamento de sistemas y computación, Tecnológico Nacional de México, Instituto Tecnológico de Toluca. México.

ABSTRACT

In the State of Mexico, as well as throughout the Mexican Republic, companies and businesses dedicated to providing construction services have had problems staying on their feet in recent years. Due to the decrease in job opportunities and the cuts in personnel, from administrative activities to those who perform services (bricklayers, architects, etc.), in the same way, applicants for construction, repair and maintenance services of buildings, when looking for someone to perform these services, they are likely to be victims of scams. And both parties are prone to informality in delivery and payment for the service.

For a student of the Computer Systems Engineering career, the development of technological skills and competences implies the use of agile technologies for the construction of applications that allow solving a real problem. In this way, a web application is created and it is this document that describes RyM as a web application created with technological development tools with a recent boom, whose objective is to reactivate an economic zone and publish the services of providers. In which applicants can contact trusted service providers and have a prior evaluation of their work.

PALABRAS CLAVE: Servicios de reparación y mantenimiento, Edificios, Aplicación web, Reactivación de zona económica.

Keywords: Maintenance and reparation services, Buildings, Web application, Reactivation of economic zone.

INTRODUCCIÓN

Recientemente las empresas que ofrecen servicios de construcción, así como las personas contratadas para la realización de dichos servicios (albañiles, plomeros, electricistas, etc.), ha experimentado una disminución en las oportunidades de encontrar obras en las cuales trabajar.

En una noticia redactada por Zavala (2021), se informa que la industria de la construcción en México tomó el tercer puesto como una de las industrias con mayores pérdidas de trabajo en el 2020, con una caída del 5.5% del personal utilizado para las labores relacionadas a dicha industria en comparación con el año 2019.

En una entrevista de AD Noticias (2020) realizada a Enrique Maza Coteró, presidente de la Cámara Mexicana de la Industria de la Construcción (CMIC), se comenta que entre 100 mil y 80 mil empleos se han perdido en toda la cadena productiva que forma parte de la industria de la construcción en el Estado de México.

Datos analizados por el Centro de Estudios del Sector de la Construcción (CEESCO), arrojan que el valor de producción de las empresas constructoras en el Estado de México fue disminuyendo drásticamente entre el año 2018 y 2021. En el 2018 se tenía una producción de alrededor de 5 mil 642 millones de pesos, la cual para el año 2021 disminuyó a 3 mil 739 millones de pesos.

Un problema habitual al que están sujetos los solicitantes de servicios son las estafas. Por ejemplo, una noticia del Sol de Yucatán (2021), así como una noticia de Impreso (2021), informan que una empresa del Estado de México, JOZUMA, ha realizado estafas a comunidades de otros estados. Con cobros excesivos o incumpliendo en los proyectos cobrados.

Otro caso de este problema es el mencionado en la noticia de Martínez (2020), Donde se menciona que, en Metepec, Estado de México, 40 inversionistas pagaron 50 millones de pesos por la construcción de una plaza comercial. Dicha construcción se debió entregar en el año 2016, y al año 2020, la construcción es inconclusa. Los inversionistas han tratado de contactar a la empresa encargada de realizar la obra sin ningún éxito.

Las herramientas emergentes para el desarrollo de software, según la institución educativa Euroinnova (s.f.) son programas informáticos que los desarrolladores de software

utilizan para crear, depurar, mantener, encontrar soluciones de errores o apoyar programas y aplicaciones.

El objetivo de esta investigación es describir el uso de herramientas emergentes; principalmente Flask, el cual es un marco de aplicaciones web de interfaz de puerta de enlace de servidor web ligero (The Pallets Projects, s.f.), y Pythonanywhere, la cual es una plataforma de alojamiento basada en la nube para el desarrollo y la implementación en línea (Mwaura, 2022); para la construcción de una aplicación web donde se publican servicios de mantenimiento en la zona conurbada Toluca y Metepec, así como mostrar el proceso de configuración, beneficios y resultados, una vez puestos en marcha sean evaluados por los usuarios.

Experimentación

La metodología utilizada para la realización de la aplicación RyM fueron las metodologías de investigación aplicada y cualitativa. Aplicativa porque se realizó una aplicación web y se compartió su enlace con algunos proveedores de servicios relacionados con la construcción de edificios, su reparación y mantenimiento, así como posibles solicitantes de dichos servicios. Cualitativa porque se realizaron encuestas para conocer la opinión de los usuarios de la aplicación, si consideran que la aplicación les puede ayudar o si la aplicación requiere más características para lograr el objetivo de RyM, además de intentar cambiar la forma tradicional de ofertar servicios y evaluar trabajos realizados, dando más certeza al usuario contratante.

MATERIAL Y MÉTODOS

Los materiales y herramientas utilizados para la elaboración de la aplicación fueron los siguientes:

- Lenguaje de marcado HTML 5.
- Lenguaje de hojas de estilo CSS.
- Lenguaje de scripts JavaScript.
- Lenguaje de programación Python.
- Micro-Framework Flask.
- Gestor de base de datos MariaDB.

- La herramienta de diseño Front-End Bootstrap 5.
- Script starr.js.
- La extensión PyMySQL 1.0.2.
- La extensión flask-socketio 4.3.1.
- La extensión flask-mail 0.9.1.
- El editor de texto Visual Studio Code.
- Pythonanywhere como host de la aplicación.
- Herramienta de diseño de diagramas Moqups.

DESARROLLO

1. Análisis de requisitos

Para la creación de una aplicación de software es necesario desarrollar el proceso de análisis de requisitos, el cual, según Pressman (2010, p. 102), proporciona el mecanismo apropiado para entender lo que desea el cliente. De este análisis se recabaron los requisitos funcionales, los cuales describen las funciones que un sistema debe realizar (García, 2015), y no funcionales, que son aquellos que engloban las restricciones y limitaciones de un sistema (García, 2015), para RyM.

Para la realización de los pre-diseños de las páginas de la aplicación se utilizó una plataforma web llamada “Moqups”. La cual es una aplicación web simplificada que te ayuda a crear y colaborar en tiempo real en esquemas, maquetas, diagramas y prototipos (Mwaura, 2022). Algunos ejemplos de estos pueden ser los diagramas UML y diseños para aplicaciones de escritorio, web y móviles.

Para utilizar esta herramienta de diagramado se tiene que crear una cuenta, para esto la plataforma solicita tu correo y una contraseña o registrarse con tu cuenta de Google. Ya registrado, puedes hacer uso de la herramienta para crear diferentes diagramas y prototipos.

Las *figuras 1 y 2* muestran un ejemplo de diagrama y de prototipo realizados con esta herramienta.

Figura 1. Ejemplo de diagrama

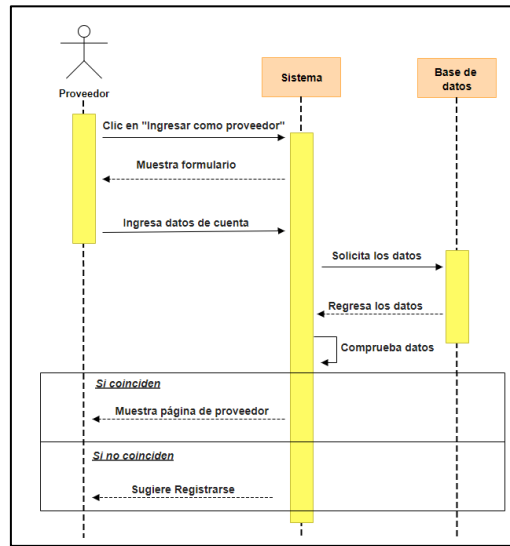


Figura 2. Ejemplo de prototipo



2. Instalación de la herramienta Flask.

Realizados los diseños de las páginas de la aplicación RyM. Se inicia la instalación de "Flask", la cual es una herramienta poco conocida. Este micro-framework utiliza el lenguaje de programación Python y cuenta con una estructura de archivos sencilla de utilizar, solo se utilizan tres carpetas esenciales para la aplicación. Un framework es una herramienta que ofrece una estructura base para la elaboración de un proyecto con objetivos específicos

(edix, 2021). Flask también cuenta con una herramienta llamada “*Jinja*”, la cual facilita la creación de los archivos HTML, ya que permite reutilizar dichos archivos como plantillas de manera sencilla y fácil de entender.

Antes de instalar “*Flask*”, el equipo donde se va a desarrollar la aplicación debe tener instalado el lenguaje de programación Python. Lo que se hace para la instalación de “*Flask*” es crear un entorno virtual dentro de una carpeta donde se almacenan todos los archivos para que la aplicación funcione correctamente, utilizando los comandos de la *figura 3* en una terminal.

Figura 3. Comandos de entorno virtual

```
# Linux
sudo apt-get install python3-venv # If needed
python3 -m venv .venv
source .venv/bin/activate

# macOS
python3 -m venv .venv
source .venv/bin/activate

# Windows
py -3 -m venv .venv
.venv\scripts\activate
```

Una vez que se ejecutaron estos comandos, y verificando que el entorno virtual esté activado, ya se puede instalar el micro-framework. Para ello se ejecuta el comando de la *figura 4* en la terminal, lo cual instalará todo lo necesario para que “*Flask*” funcione correctamente.

Figura 4. Comando instalación de Flask

```
python -m pip install flask
```

En relación con las carpetas antes mencionadas; la estructura que deben seguir es la siguiente: la carpeta donde se guardan los archivos para que el entorno pueda ejecutar la aplicación, en el caso de RyM es la carpeta de entorno virtual, la cual se llamó “*venv*”; la carpeta denominada “*templates*”, donde se mantienen los documentos HTML de la aplicación; y la carpeta denominada “*static*”, donde se guardan archivos varios, en el caso de RyM, en esta carpeta se mantienen los archivos *.js*, *.css* y las imágenes subidas por los

usuarios de la aplicación. También, es necesario tener un archivo `.py` el cual es la aplicación. En ese solo archivo se puede tener toda la aplicación si se quiere, por esa razón se le conoce a “*Flask*” como un micro-framework.

En el caso de RyM, se utilizaron varios archivos `.py`. El archivo `app.py`, el cual realiza la carga de las páginas de la aplicación, así como la dirección y redirección a través de ellas, y el llamado de los diferentes métodos de los demás archivos; el archivo `utilidades` en el cual se tiene métodos varios para la funcionalidad de la aplicación; y los archivos `conexión_bd.py` y `controlador.py`, el cual contiene los métodos para hacer la conexión con la base de datos y la modificación, registro y eliminación de información dentro de la base de datos.

Una vez que se generó la estructura de archivos y carpetas, se verifica que el entorno virtual esté activado, si no lo está, se debe ejecutar el comando que se asemeja a una ruta de archivo, mostrado en la *fig. 1*. Ya que se verificó que el entorno virtual está activado, se instalan las demás herramientas que ofrece “*Flask*” y Python.

3. Instalación y conexión con bases de datos

Para realizar la inserción, modificación y eliminación de información dentro de la base de datos se requiere de la extensión *PyMySQL*, la cual forma parte de las herramientas disponibles al descargar el lenguaje de programación Python.

Para instalar la extensión “*PyMySQL*” se ejecuta el comando: ***pip install PyMySQL*** en la terminal. La *figura 5* muestra la correcta instalación de la extensión.

Figura 5. Ejemplo instalación de PyMySQL

```
PS D:\Documentos\Respaldo_OMEN\Tec\Proyecto_RyM> .\venv\Scripts\activate
(venv) PS D:\Documentos\Respaldo_OMEN\Tec\Proyecto_RyM> pip install PyMySQL
Collecting PyMySQL
  Using cached PyMySQL-1.0.2-py3-none-any.whl (43 kB)
Installing collected packages: PyMySQL
Successfully installed PyMySQL-1.0.2
```

Instalada la extensión, lo único que se tiene que hacer es importar el archivo `.py`, instalado al ejecutar el comando, en el archivo `.py` donde se desarrollarán los métodos de conexión y modificación de información dentro de la base de datos.

4. Instalación de herramienta SMTP

Según IBM (2021), el protocolo simple de transferencia de correos (SMTP) es un protocolo TCP/IP utilizado para enviar y recibir correos electrónicos. Por lo que se necesita de un servidor SMTP para que una aplicación pueda enviar correos.

Para que una aplicación web elaborada con Flask pueda enviar correos es necesario instalar la extensión “*flask-mail*”. Para instalarla, se ejecuta el comando: ***pip install flask-mail*** en la terminal. La *figura 6* ejemplifica la correcta instalación de la extensión.

Figura 6. Ejemplo instalación de flask-mail

```
PS D:\documentos\Respaldo_OMEN\Tec\Proyecto_RyM> & d:/documentos/Respaldo_OMEN/Tec/Proyecto_RyM/venv/Scripts/Activate.ps1
(venv) PS D:\documentos\Respaldo_OMEN\Tec\Proyecto_RyM> pip install flask-mail
Collecting flask-mail
  Downloading Flask-Mail-0.9.1.tar.gz (45 kB)
    45.7/45.7 KB 2.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: Flask in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from flask-mail) (2.0.2)
Collecting blinker
  Downloading blinker-1.4.tar.gz (111 kB)
    111.5/111.5 KB 1.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: click>=7.1.2 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from Flask->flask-mail) (8.0.3)
Requirement already satisfied: Jinja2>=3.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from Flask->flask-mail) (3.0.3)
Requirement already satisfied: Werkzeug>=2.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from Flask->flask-mail) (2.0.2)
Requirement already satisfied: itsdangerous>=2.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from Flask->flask-mail) (2.0.1)
Requirement already satisfied: colorama in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from click>=7.1.2->Flask->flask-mail) (0.4.4)
Requirement already satisfied: MarkupSafe>=2.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from Jinja2>=3.0->Flask->flask-mail) (2.0.1)
Using legacy 'setup.py install' for flask-mail, since package 'wheel' is not installed.
Using legacy 'setup.py install' for blinker, since package 'wheel' is not installed.
Installing collected packages: blinker, flask-mail
  Running setup.py install for blinker ... done
  Running setup.py install for flask-mail ... done
Successfully installed blinker-1.4 flask-mail-0.9.1
```

Con esta extensión instalada, se necesita configurar los parámetros necesarios para que, al momento de llamar al método encargado de enviar los correos, la aplicación identifique que servidor SMTP a utilizar. La configuración de los parámetros se puede realizar directamente desde el archivo *.py* donde se tenga la aplicación o desde otro archivo. Este último caso fue el utilizado para RyM.

Los parámetros por configurar son:

- El servidor SMTP.
- El puerto.
- El protocolo de seguridad a utilizar.
- El nombre de usuario del correo.
- La contraseña del usuario.

Con estos parámetros indicados lo restante es cargar la configuración en la aplicación para que el método de envío de correos funcione correctamente.

5. Instalación de herramienta de comunicación bidireccional en tiempo real.

Para que en una aplicación desarrollada con *Flask* cuente con una comunicación bidireccional basada en eventos en tiempo real, se requiere instalar la extensión “*flask-socketIO*”. La extensión instala tres elementos que, según el escritor de la documentación Grinberg (2018), son utilizados como protocolo de transporte que permite la comunicación bidireccional a través de eventos en tiempo real entre el cliente (generalmente, navegadores web) y un servidor.

Para su instalación, se ejecuta el comando: ***pip install --upgrade flask-socketio==4.3.1*** en la terminal. Instalada esta extensión se tendrá que cambiar la versión de las otras herramientas que se instalaron al ejecutar el comando, debido a que existe una incompatibilidad entre las versiones. Esta incompatibilidad hace que una de las funciones trabaje de manera incorrecta. Para las otras herramientas se tienen que ejecutar los comandos: ***pip install --upgrade python-engineio==3.13.2*** y ***pip install --upgrade python-socketio==4.6.0***. La *figura. 7* y *figura 8* muestran la instalación con las versiones.

Figura 7. Instalación flask-socketio 4.3.1

```
(venv) PS D:\Documentos\Respaldo_OMEN\Tec\Proyecto_RyM> pip install --upgrade flask-socketio==4.3.1
Collecting flask-socketio==4.3.1
  Using cached Flask_SocketIO-4.3.1-py2.py3-none-any.whl (15 kB)
Collecting python-socketio>=4.3.0
  Using cached python_socketio-5.6.0-py3-none-any.whl (56 kB)
Requirement already satisfied: Flask>=0.9 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from flask-socketio==4.3.1) (2.0.2)
Requirement already satisfied: itsdangerous>=2.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from Flask>=0.9->flask-socketio==4.3.1) (2.0.1)
Requirement already satisfied: Jinja2>=3.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from Flask>=0.9->flask-socketio==4.3.1) (3.0.3)
Requirement already satisfied: click>=7.1.2 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from Flask>=0.9->flask-socketio==4.3.1) (8.0.3)
Requirement already satisfied: Werkzeug>=2.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from Flask>=0.9->flask-socketio==4.3.1) (2.0.2)
Requirement already satisfied: bidict>=0.21.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from python-socketio>=4.3.0->flask-socketio==4.3.1) (0.22.0)
Collecting python-engineio>=4.3.0
  Using cached python_engineio-4.3.2-py3-none-any.whl (52 kB)
Requirement already satisfied: MarkupSafe>=2.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-packages (from Jinja2>=3.0->Flask>=0.9->flask-socketio==4.3.1) (2.0.1)
Installing collected packages: python-engineio, python-socketio, flask-socketio
Successfully installed flask-socketio-4.3.1 python-engineio-4.3.2 python-socketio-5.6.0
```

Figura 8. Instalación python-engineio 3.13.2 y python-socketio 4.6.0

```
(venv) PS D:\Documentos\Respaldo_OMEN\Tec\Proyecto_RyM> pip install --upgrade python-engineio==3.13.2
Collecting python-engineio==3.13.2
  Using cached python_engineio-3.13.2-py2.py3-none-any.whl (50 kB)
Requirement already satisfied: six>=1.9.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-p
ackages (from python-engineio==3.13.2) (1.16.0)
Installing collected packages: python-engineio
  Attempting uninstall: python-engineio
    Found existing installation: python-engineio 4.3.2
    Uninstalling python-engineio-4.3.2:
      ERROR: pip's dependency resolver does not currently take into account all the packages that are installed
. This behaviour is the source of the following dependency conflicts.
python-socketio 5.6.0 requires python-engineio>=4.3.0, but you have python-engineio 3.13.2 which is incom
patible.
Successfully installed python-engineio-3.13.2
(venv) PS D:\Documentos\Respaldo_OMEN\Tec\Proyecto_RyM> pip install --upgrade python-socketio==4.6.0
Collecting python-socketio==4.6.0
  Downloading python_socketio-4.6.0-py2.py3-none-any.whl (51 kB)
    51.8/51.8 KB 2.8 MB/s eta 0:00:00
Requirement already satisfied: python-engineio>=3.13.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\ve
nv\lib\site-packages (from python-socketio==4.6.0) (3.13.2)
Requirement already satisfied: six>=1.9.0 in d:\documentos\respaldo_omen\tec\proyecto_rym\venv\lib\site-p
ackages (from python-socketio==4.6.0) (1.16.0)
Installing collected packages: python-socketio
  Attempting uninstall: python-socketio
    Found existing installation: python-socketio 5.6.0
    Uninstalling python-socketio-5.6.0:
      Successfully uninstalled python-socketio-5.6.0
Successfully installed python-socketio-4.6.0
(venv) PS D:\Documentos\Respaldo_OMEN\Tec\Proyecto_RyM> █
```

Instaladas estas herramientas y al contar con la base de datos para pruebas lista, se empieza con la programación de la aplicación RyM.

6. Programación de la aplicación y pruebas

Lo esencial de la programación es instanciar el objeto *app* y determinar los enlaces de la aplicación, así como las acciones que se realizarán al momento de ingresar en dichos enlaces dentro del archivo *.py* seleccionado para ser la aplicación web.

Para instanciar el objeto *app* es necesario importar la clase *Flask* dentro del archivo *.py*, seguido de esto, se instancia el objeto *app*, el cual se utiliza para crear las rutas de enlace de la aplicación web. La *fig. 9* lo mencionado.

Figura 9. Instancia del objeto *app*

```
from flask import Flask

app = Flask(__name__)
```

Para determinar los enlaces se requiere seguir la estructura de las instrucciones mostradas en la *figura 10*. En primer lugar, se utiliza la línea *@app.route("{ruta}")* para indicar la ruta del enlace, en el caso de la figura es la página principal de RyM. Después de

esa línea se crea una función la cual será llamada al momento de escribir la ruta en la barra de navegación de un explorador y dentro de ella se colocan las acciones deseadas, en el ejemplo de la figura, se verifica si el usuario inicio sesión como proveedor, solicitante o ninguno, y al hacer clic en el logo en la barra superior de navegación, se le dirige a la página principal para proveedores, solicitantes o a la de inicio de RyM. Esto se realiza con la función *render_template* ('{nombre del archivo html}') o utilizando la anidación de las funciones *redirect* (*url_for* ('{nombre de la función}')).

Figura 10. Ejemplo de la creación de rutas

```
@app.route("/")
def index():
    if "id_usuarioP" in session:
        return redirect(url_for("inicio_proveedor"))
    elif "id_usuarioC" in session:
        return redirect(url_for("inicio_cliente"))
    else:
        return render_template('index.html')
```

Lo importante a destacar respecto a la estructuración y diseño de los archivos HTML es la herramienta *Jinja*, el cual es un motor de plantillas rápido, expresivo y extensible que utiliza una sintaxis similar a la de Python (Pallets, 2007). *Jinja* viene como un elemento extra al momento de instalar *Flask*.

Su utilización es sencilla. Para crear una plantilla lo que se tiene que hacer es crear un archivo HTML, colocar las etiquetas de los elementos que se repetirán en los demás archivos HTML y crear bloques. Estos son una de las bondades de *Jinja*. Para crear un bloque se tiene que seguir la siguiente sintaxis: *{% block nombre del bloque %} {% end block %}*. La *fig. 11* muestra un ejemplo de lo mencionado.

Figura 11. Ejemplo de plantilla

```
15 <div class="container-fluid">
16   <header class="row">
17     <nav class="navbar navbar-expand-lg navbar-dark es-navbar">
18       <div class="container-fluid">
19         <a class="navbar-brand" href="{{url_for('index')}}"><img class="img-logo"
20           <button class="navbar-toggler" type="button" data-bs-toggle="collapse" da
21             <span class="navbar-toggler-icon"></span>
22           </button>
23           <div class="collapse navbar-collapse" style="margin-right: auto;" id="nav
24             {% block sesion %}{% endblock %}
25           </div>
26         </div>
27       </nav>
28     </header>
29     <body>
30       <main>
31         {% block mensajes_flash %}{% endblock %}
32         {% block cuerpo%}{% endblock %}
33       </main>
34     </body>
```

Para utilizar la plantilla se requiere importarla a un archivo HTML que la requiera. Esto se realiza con la siguiente sintaxis: `{% extends 'nombre de la plantilla.html' %}`. Después, se deben colocar los bloques a utilizar y dentro de estos poner las etiquetas de los elementos exclusivos del archivo HTML. La *figura 12* ejemplifica la importación y el llamado de bloques.

Figura 12. Ejemplo importación y bloques de plantilla

```
1  {% extends 'layout.html' %}
2
3  {% block titulo %}Crea tu cuenta de proveedor{% endblock %}
4  {% block imports_extras %}
5  <script src="../../static/js/validacion_registroP.js"></script>
6  {% endblock %}
7
8  {% block mensajes_flash %}
9    {% with messages = get_flashed_messages() %}
10     {% if messages %}
11       {% for message in messages %}
12         <div class="alert alert-danger alert-dismissable">
13           <p class="fs-5">{{ message }}</p>
14           <button type="button" class="btn-close"
15             </div>
16         {% endfor %}
17       {% endif %}
18     {% endwith %}
19   {% endblock %}
20
21   {% block cuerpo %}
22   <form action="/registro_proveedor" id="formulario_registro" clas
23     <div class="row g-2">
```

Otra bondad de *Jinja* es que permite la utilización de la sentencia condicional *if* y la sentencia de repetición *for*, como se muestra en la *fig. 12*. Esto permite que el servidor

agregue elementos en el archivo HTML de manera condicional o dinámica. Para las pruebas de RyM se realizaron pruebas de caja negra, el cual es un método de prueba de software en el que las funcionalidades de la aplicación se prueban sin tener conocimiento de la estructura del código interno, los detalles de implementación y las rutinas internas (Hamilton, 2022).

Para los solicitantes de servicios se probaron las funciones con el siguiente orden:

- 1) *Creación de cuenta*. En esta prueba se verificó el funcionamiento de la validación de campos y que los datos se almacenaran correctamente en la base de datos.
- 2) *Modificación de información de cuenta*. En esta prueba se verificó que la validación de campos funcionara correctamente, que la información recolectada de la base de datos se mostrara y que los datos dentro de la base de datos se modificaran correctamente.
- 3) *Visualización de servicios*. Se inspeccionó que la información obtenida de la base de datos acerca de los servicios fuera correcta y que dicha información se mostrara en la página de servicios.
- 4) *Visualización de proveedores*. Se aseguró que la información recolectada de la base de datos acerca de los proveedores fuera correcta y que se visualizara en la página de proveedores.
- 5) *Dejar un comentario*. Se verificó que el comentario escrito se almacenara en la base de datos.
- 6) *Solicitud de datos de contacto*. Se inspeccionó que la información de contacto del proveedor se recuperara y que se enviara al correo del solicitante correctamente.
- 7) *Solicitud de servicio*. Se verificó la validación de los campos y que la información escrita en ellos se enviara al correo del proveedor adecuadamente.
- 8) *Calificar servicio*. Se verificó que la calificación se almacenara en la base de datos.

Para los proveedores de servicios se probaron las funciones con el siguiente orden:

- 1) *Creación de cuenta*. Se valida que los campos respeten las condiciones establecidas para almacenar los datos y que se almacenen correctamente en la base de datos.

- 2) *Modificación de información de cuenta.* Se inspeccionó que la validación de campos funcionara adecuadamente y que la información almacenada dentro de la base de datos se modificara.
- 3) *Registro de servicios.* Se verificó que la validación de campos funcionara correctamente y que la información se registrara en la base de datos.
- 4) *Modificación de información de servicios.* Se aseguró que la validación de campos funcionara, que la información recuperada de la base de datos se mostrara y que dicha información se modificara en la base de datos.

7. Pythonanywhere para crear el host

Al finalizar la programación y las pruebas se procede a subir la aplicación a Pythonanywhere, la cual es una aplicación que aloja páginas y aplicaciones web hechas con un framework que utiliza el lenguaje de programación Python. Para subir la aplicación se creó un repositorio en GitHub con los archivos de la aplicación RyM. Después es necesario crear una cuenta en Pythonanywhere.

Al tener la cuenta creada y navegando por la página se puede encontrar una pestaña para crear una consola, en ella se clona el repositorio de GitHub creado anteriormente y se crea el entorno virtual que fungirá como el entorno de producción. Al igual que en el entorno creado en el desarrollo, se tiene que instalar las extensiones necesarias. La *figura 13* ejemplifica esta acción.

Figura 13. Instalación de entorno de producción

```
Bash console 24239429
19:06 ~ | ls
19:06 ~ | git clone https://github.com/Jonathan-AV/rym.git
Cloning into 'rym'...
Username for 'https://github.com': Jonathan-AV
Password for 'https://github.com':
remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.
fatal: Authentication failed for 'https://github.com/Jonathan-AV/rym.git/'
19:07 ~ | git clone https://github.com/Jonathan-AV/rym.git
Cloning into 'rym'...
Username for 'https://github.com': Jonathan-AV
Password for 'https://Jonathan-AV@github.com':
remote: Enumerating objects: 2157, done.
remote: Counting objects: 100% (2157/2157), done.
remote: Compressing objects: 100% (390/390), done.
remote: Total 2157 (delta 162), reused 2157 (delta 162), pack-reused 0
Receiving objects: 100% (2157/2157), 8.49 MiB | 24.48 MiB/s, done.
Resolving deltas: 100% (162/162), done.
Updating files: 100% (2004/2004), done.
19:14 ~ | ls
19:14 ~ | cd rym
19:15 ~/rym (main) | ls
-pycache_ app.py conexion_bd.py config.py controlador.py requirements.txt static templates utilidadesRYM.py venv
19:15 ~/rym (main) | ls
-pycache_ app.py conexion_bd.py config.py controlador.py requirements.txt static templates utilidadesRYM.py
19:20 ~/rym (main) | virtualenv --python=python3.10 venv
RuntimeError: failed to find interpreter for Builtin discover of python_spec='python3.10'
19:21 ~/rym (main) | virtualenv --python=python3.9 venv
created virtual environment CPython3.9.5.final.0-64 in 12525ms
creator CPython3Posix(dest=/home/RYM/rym/venv, clear=False, no_vcs_ignore=False, global=False)
sender framework(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via-copy, app_data_dir=/home/RYM/.local/share/virtualenv)
added seed packages: pip=21.1.2, setuptools=57.0.0, wheel=0.38.2
activators BashActivator,ShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
19:26 ~/rym (main) | ls
-pycache_ app.py conexion_bd.py config.py controlador.py requirements.txt static templates utilidadesRYM.py venv
19:26 ~/rym (main) | source venv/bin/activate
(venv) 19:27 ~/rym (main) | pip install flask
looking in links: /usr/share/pip-wheels
Collecting flask
  Downloading Flask-2.1.2-py3-none-any.whl (95 kB)
Collecting itsdangerous==2.0
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting werkzeug==2.0
  Downloading werkzeug-2.1.2-py3-none-any.whl (224 kB)
Collecting importlib-metadata==3.6.0
  Downloading importlib_metadata-4.11.3-py3-none-any.whl (18 kB)
Collecting click==8.0
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
Collecting Jinja2==3.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
Collecting zipp==0.3
  Downloading zipp-3.8.0-py3-none-any.whl (5.4 kB)
Collecting MarkupSafe==2.0
  Downloading MarkupSafe-2.1.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
```

Teniendo el entorno creado, es necesario crear la base de datos que guardará los datos reales de los usuarios de la aplicación. Para ello se abre una consola de MySQL en Pythonanywhere, y se colocan todas las instrucciones de creación de tablas, relaciones y funciones necesarias para que la aplicación funcione correctamente. La figura 14 muestra esta acción.

Figura 14. Base de datos de producción

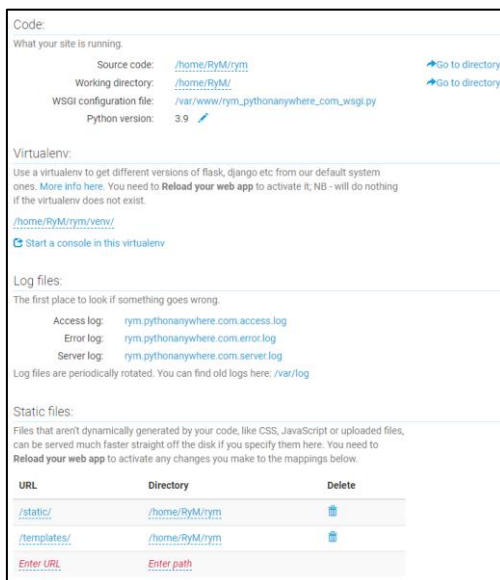
```
MySQL: RYMDefault
Database
information_schema
RYMDefault
2 rows in set (1.32 sec)

mysql> use RYMDefault
Database changed
mysql> CREATE TABLE `cliente` (
  -> `id_cliente` int(11) NOT NULL AUTO_INCREMENT,
  -> `nombre` varchar(35) COLLATE utf8_spanish_ci NOT NULL,
  -> `ape_paterno` varchar(35) COLLATE utf8_spanish_ci NOT NULL,
  -> `ape_materno` varchar(35) COLLATE utf8_spanish_ci NOT NULL,
  -> `telefono` varchar(10) COLLATE utf8_spanish_ci NOT NULL,
  -> `correo` text COLLATE utf8_spanish_ci NOT NULL,
  -> `contrasena` text COLLATE utf8_spanish_ci NOT NULL,
  -> `img_perfil` text COLLATE utf8_spanish_ci DEFAULT NULL,
  -> PRIMARY KEY (`id_cliente`),
  -> UNIQUE KEY `correo` (`correo`) USING HASH
  -> ) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
ERROR 1170 (42000): BLOB/TEXT column 'correo' used in key specification without a key length
mysql> CREATE TABLE `cliente` (
  -> `id_cliente` int(11) NOT NULL AUTO_INCREMENT,
  -> `nombre` varchar(35) COLLATE utf8_spanish_ci NOT NULL,
  -> `ape_paterno` varchar(35) COLLATE utf8_spanish_ci NOT NULL,
  -> `ape_materno` varchar(35) COLLATE utf8_spanish_ci NOT NULL,
  -> `telefono` varchar(10) COLLATE utf8_spanish_ci NOT NULL,
  -> `correo` text(150) COLLATE utf8_spanish_ci NOT NULL,
  -> `contrasena` text COLLATE utf8_spanish_ci NOT NULL,
  -> `img_perfil` text COLLATE utf8_spanish_ci DEFAULT NULL,
  -> PRIMARY KEY (`id_cliente`),
  -> UNIQUE KEY `correo` (`correo`) USING HASH
  -> ) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
ERROR 1170 (42000): BLOB/TEXT column 'correo' used in key specification without a key length
mysql> CREATE TABLE `cliente` (
  -> `id_cliente` int(11) NOT NULL AUTO_INCREMENT,
  -> `nombre` varchar(35) COLLATE utf8_spanish_ci NOT NULL,
  -> `ape_paterno` varchar(35) COLLATE utf8_spanish_ci NOT NULL,
  -> `ape_materno` varchar(35) COLLATE utf8_spanish_ci NOT NULL,
  -> `telefono` varchar(10) COLLATE utf8_spanish_ci NOT NULL,
  -> `correo` varchar(200) COLLATE utf8_spanish_ci NOT NULL,
  -> `contrasena` text COLLATE utf8_spanish_ci NOT NULL,
  -> `img_perfil` text COLLATE utf8_spanish_ci DEFAULT NULL,
  -> PRIMARY KEY (`id_cliente`),
  -> UNIQUE KEY `correo` (`correo`) USING HASH
  -> ) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE `casi_servicios` (
  -> `id_servicio` int(11) NOT NULL,
  -> `id_cliente` int(11) NOT NULL,
  -> `calificacion` int(11) NOT NULL,
  -> KEY `fk_id_cliente_casi` (`id_cliente`),
  -> KEY `fk_id_servicio_casi` (`id_servicio`),
  -> CONSTRAINT `fk_id_cliente_casi` FOREIGN KEY (`id_cliente`) REFERENCES `cliente` (`id_cliente`) ON DELETE CASCADE ON UPDATE CASCADE,
  -> CONSTRAINT `fk_id_servicio_casi` FOREIGN KEY (`id_servicio`) REFERENCES `servicio` (`id_servicio`) ON DELETE CASCADE ON UPDATE CASCADE,
  -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
ERROR 1215 (HY000): Cannot add foreign key constraint
```

Contando con estas acciones realizadas, en Pythonanywhere es necesario dirigirse a su apartado “web”. Aquí se configuran las rutas del archivo *app.py*, del entorno virtual y las carpetas *static* y *templates*. Para que Pythonanywhere tenga identificado donde está cada archivo al que se le hace referencia en el archivo *app.py*. La *figura 15* ejemplifica esta configuración.

Figura 15. Configuración de rutas en Pythonanywhere



RESULTADOS

La aplicación creada se colocó en el hosting Pythonanywhere. La cual al configurar todo lo que se realizó en el apartado de “Desarrollo” genera un enlace para acceder a la aplicación: <http://rym.pythonanywhere.com/>

Dentro de las funciones desarrolladas más llamativas de la aplicación RyM se encuentran que los solicitantes pueden visualizar las imágenes de los servicios realizados por los proveedores, calificar los servicios, solicitar servicios al proveedor, solicitar sus datos de contacto y dejar comentarios. Mientras que los proveedores pueden registrar uno o varios servicios, así como modificar la información del servicio registrado y agregar y eliminar imágenes de dicho servicio. La *fig. 15* muestra el registro de servicios y la *figura 16* muestra la modificación de la información de los servicios.

Secciones de la aplicación RyM

1. Página principal de RyM (figura 16).

Figura 16. Página principal de la aplicación



2. Registro de proveedor (figura. 17)

Figura 17. Registro de proveedor

The screenshot shows the 'Registro de proveedor' form. It has a teal header with the RyM logo. The form is divided into two main sections: 'Registro de información:' and 'Creación de cuenta:'. Under 'Registro de información:', there are four input fields: 'Nombre (necesario)', 'Apellido Paterno (necesario)', 'Apellido Materno (necesario)', and 'Oficio o Giro de empresa (necesario)'. There are also two optional fields: 'Dirección de su empresa (opcional)' and 'Teléfono (necesario)'. Under 'Creación de cuenta:', there are three input fields: 'Correo Electrónico (necesario)', 'Contraseña (necesario)', and 'Repita su contraseña (necesario)'. Below these fields is a checkbox labeled 'Acepto los términos y condiciones.' and a teal button labeled 'Crear cuenta'.

3. Búsqueda de servicio (figura 18)

Figura 18. Búsqueda de servicio



4. Búsqueda de proveedor (fig. 19)

Figura 19. Búsqueda de proveedores



5. Solicitud de servicio, evaluación de servicio y comentarios.

La figura 20 muestra la página donde los solicitantes de servicios realizan la mayoría de las funciones.

Figura 20. Funciones para solicitantes de servicios



6. Modificación de información de cuenta de proveedor (figura 21), de solicitante (figura 22) y modificación de la información de un servicio (figura 23).

Figura 21. Modificar información (proveedor)

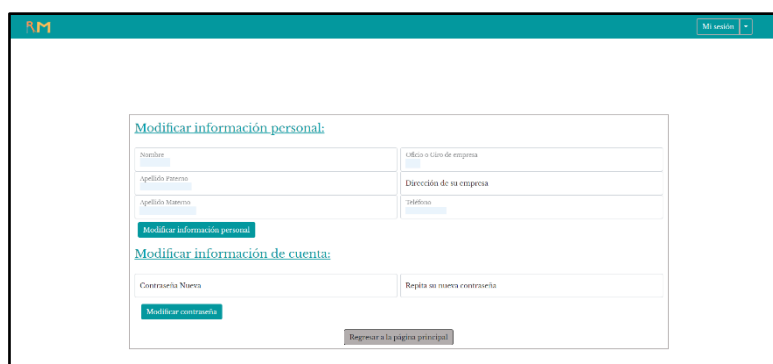


Figura 22. Modificar información (solicitante)

The screenshot shows a web interface for modifying user information. It is divided into two main sections:

- Modificar información personal:** This section contains input fields for 'Nombre', 'Apellido Paterno', 'Apellido Materno', and 'Teléfono'. Below these fields is a green button labeled 'Modificar información personal'.
- Modificar información de cuenta:** This section contains two input fields for 'Contraseña Nueva' and 'Repita su nueva contraseña'. Below these is a green button labeled 'Modificar contraseña'.

At the bottom of the form, there is a section for profile picture changes: 'Cambiar imagen de perfil' with a 'Seleccionar archivo' button and a 'Ninguno archivo selet.' message. Below this is a green button labeled 'Modificar imagen de perfil'. At the very bottom center is a grey button labeled 'Regresar a la página principal'.

Figura 23.- Modificación de la información de un servicio.

The screenshot shows a web interface for modifying service information. It includes:

- A header with the RyM logo and 'Mi sesión'.
- A section titled 'Modificar la información del servicio' with a dropdown menu for 'Elegir el servicio' and a green button labeled 'Modificar servicio'.
- A section titled 'Modifique sus imágenes del servicio' with two image thumbnails and green buttons labeled 'Eliminar Imagen'.
- A section titled 'Agregue más fotos de su servicio' with a 'Elegir archivos' button and a 'Ningunos archivos selet.' message, and a green button labeled 'Agregar Imagen'.
- A grey button labeled 'Regresar a las sesiones' at the bottom center.

Evaluación de plataforma.

A través de una encuesta se verificó el grado de satisfacción que tuvieron tanto los solicitantes de servicios como los proveedores de servicios. En esta encuesta los resultados más relevantes son; por parte de los solicitantes, que el diseño de la aplicación debe ser más atractiva y que se requiere más información de los proveedores de servicios para que sean de confianza; y por parte de los proveedores de servicios, se mencionó que también se debe tomar en cuenta a las empresas, negocios y personas morales para el registro como proveedor en RyM.

DISCUSIONES Y CONCLUSIONES

Las herramientas emergentes Flask y Pythonanywhere tienen una curva de aprendizaje rápida, no solo por la sencillez y comprensión fácil de la documentación, sino que también se cuentan con otros recursos que pueden facilitar la comprensión y el uso de estas herramientas. Estas también son de gran utilidad si se está aprendiendo a realizar el Front-End de una aplicación web, ya que al utilizar Python la programación es más sencilla gracias a las diferentes bondades que ofrece dicho lenguaje; y porque te facilita el probar alguna aplicación desarrollada en un entorno de producción y poder compartir tu desarrollo con más personas sin la complejidad que conlleva el hacer un servidor propio, generar un dominio, etc.

RyM es una buena opción para que pequeños negocios, así como individuos, publiciten sus servicios. Ya que les podrán solicitar servicios y obtener un ingreso extra mientras están a la espera de ser contratados para la construcción o remodelación total o parcial de algún edificio; y para las personas que quieran solicitar es un medio refrescante por el cual pueden contactar con los proveedores de servicios.

Pero algo es claro, se debe trabajar en un diseño más agradable, con una estructura de colores mejor formada y con más elementos dinámicos (animaciones); identificar otras características o información necesaria para que a más solicitantes de servicios les genere confianza la aplicación y por ende que todos los proveedores que publican sus servicios en ella sean considerados de confianza; y no solo considerar a los pequeños negocios e individuos, también a negocios o empresas más grandes. Esto último es un gran acierto, porque por lo general los albañiles son contratados por algún negocio o empresa para realizar un servicio. Rara es la ocasión en la que un albañil busca por su cuenta alguna obra en la cual trabajar.

Para finalizar, si se desea desarrollar alguna aplicación, verificar la compatibilidad de las diferentes herramientas que se van a utilizar en el desarrollo es algo muy esencial. En especial, si es la primera vez que se empieza a desarrollar algo propio que se desea compartir con el mundo, porque esto puede retrasar el desarrollo y modificar las fechas de entrega ya planeadas dentro de algún cronograma de actividades.

REFERENCIAS BIBLIOGRÁFICAS

AD Noticias. (noviembre 26, 2020). Pierde más de 80 mil empleos la construcción en el Edomex. Fecha de recuperación: febrero 10, 2020. Recuperado de: <https://adnoticias.mx/pierde-mas-de-80-mil-empleos-la-construccion-en-el-edomex/>

CEESCO. (octubre 15, 2021). PRODUCCIÓN DE LAS EMPRESAS CONSTRUCTORAS EN MÉXICO. Fecha de recuperación: febrero 11, 2022. Recuperado de: <https://www.cmic.org.mx/ceesco/produccionempresas/Informe%20Estatal%20-%20M%C3%A9xico%202021.pdf>

Edix. (agosto 19, 2021). Framework. Fecha de recuperación: junio 21, 2022. Recuperado de: <https://www.edix.com/es/instituto/framework/>

Eurounnova. (s.f.). Características de las herramientas de programación. Fecha de recuperación: junio 17, 2022. Recuperado de: <https://www.euroinnova.mx/blog/herramientas-de-programacion#caracteriacutesticas-de-las-herramientas-de-programacioacuten>

García Notario, D. (octubre 5, 2015). Análisis de requisitos en el desarrollo del software. Universidad Carlos III de Madrid. Fecha de recuperación: junio 17, 2022. Recuperado de: https://e-archivo.uc3m.es/bitstream/handle/10016/26008/PFC_Dalila_Garcia_Notario.pdf

Grinberg, M. (2018). What is Engine.IO?. Fecha de recuperación: junio 19, 2022. Recuperado de: <https://python-engineio.readthedocs.io/en/latest/intro.html#what-is-engine-io>

Grinberg, M. (2018). What is Socket.IO?. Fecha de recuperación: junio 19, 2022. Recuperado de: <https://python-socketio.readthedocs.io/en/latest/intro.html#what-is-socket-io>

Hamilton, T. (abril 30, 2022), What is BLACK Box Testing? Techniques, Example & Types. Guru99. Fecha de recuperación: junio 20, 2022. Recuperado de: <https://www.guru99.com/black-box-testing.html>

IBM. (abril 14, 2021). Protocolo simple de transferencia de correo (SMTP). Fecha de recuperación: junio 19, 2022. Recuperado de: <https://www.ibm.com/docs/es/i/7.3?topic=information-smtp>

Impreso. (febrero 8, 2021). Exigen les devuelvan \$679,000. Diario de Yucatán. Fecha de recuperación: febrero 13, 2022. Recuperado de: <https://www.yucatan.com.mx/yucatan/2021/2/8/exigen-les-devuelvan-679000-236876.html>

Martínez, A. (diciembre 21, 2020). Inversionistas fueron defraudados con más de 50 millones de pesos en Metepec. MILENIO. Fecha de recuperación: febrero 13, 2022. Recuperado de: <https://www.milenio.com/politica/comunidad/metepec-inversionistas-fueron-defraudados-por-mas-de-50-millones>

Mwaura, W. (abril 25, 2022). Automating Flask deployments with PythonAnywhere. Circleci Blog. Fecha de recuperación: junio 17, 2022. Recuperado de: <https://circleci.com/blog/automating-flask-deployments-with-pythonanywhere/>

Pallets. (2007). Jinja. Fecha de recuperación: junio 20, 2022. Recuperado de: <https://jinja.palletsprojects.com/en/3.1.x/>

Pressman, R. S. (2010). INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO. (7ma ed.). McGrawHill Educación. ISBN: 978-607-15-0314-5

Sol de Yucatán. (enero 5, 2021). EMPRESA DE EDOMEX QUIERE REPETIR EL FRAUDE EN YUCATÁN CON PIES DE CASA. Fecha de revisión: febrero 13, 2022. Recuperado de: <https://solyucatan.mx/empresa-de-edomex-quiere-repetir-el-fraude-en-yucatan-con-pies-de-casa/>

The Pallets Projects. (s.f.). Flask. Fecha de recuperación: junio 17, 2022. Recuperado de: <https://palletsprojects.com/p/flask/>

Zavala, D. (enero 26, 2021). El desempleo arrastró a la industria de la construcción. Obras por expansión. Fecha de recuperación: septiembre 17, 2021. Recuperado de: <https://obras.expansion.mx/construccion/2021/01/26/el-desempleo-arrastro-a-la-industria-de-la-construccion>