



ProfesorenVideo.com

# Curso de Python

## Aplicación monolítica con Flask - (**app.py**)

```
import ...  
import ...
```

```
registrar_alumnos():
```

```
.....
```

```
consultar_alumnos():
```

```
.....
```

```
modificar_alumnos():
```

```
.....
```

```
listar_alumnos():
```

```
.....
```

```
registrar_cursos():
```

```
.....
```

```
consultar_cursos():
```

```
.....
```

```
modificar_cursos():
```

```
.....
```

```
listar_cursos():
```

```
.....
```

```
registrar_profesores():
```

```
.....
```

```
consultar_profesores():
```

```
.....
```

```
modificar_profesores():
```

```
.....
```

```
listar_profesores():
```

```
.....
```

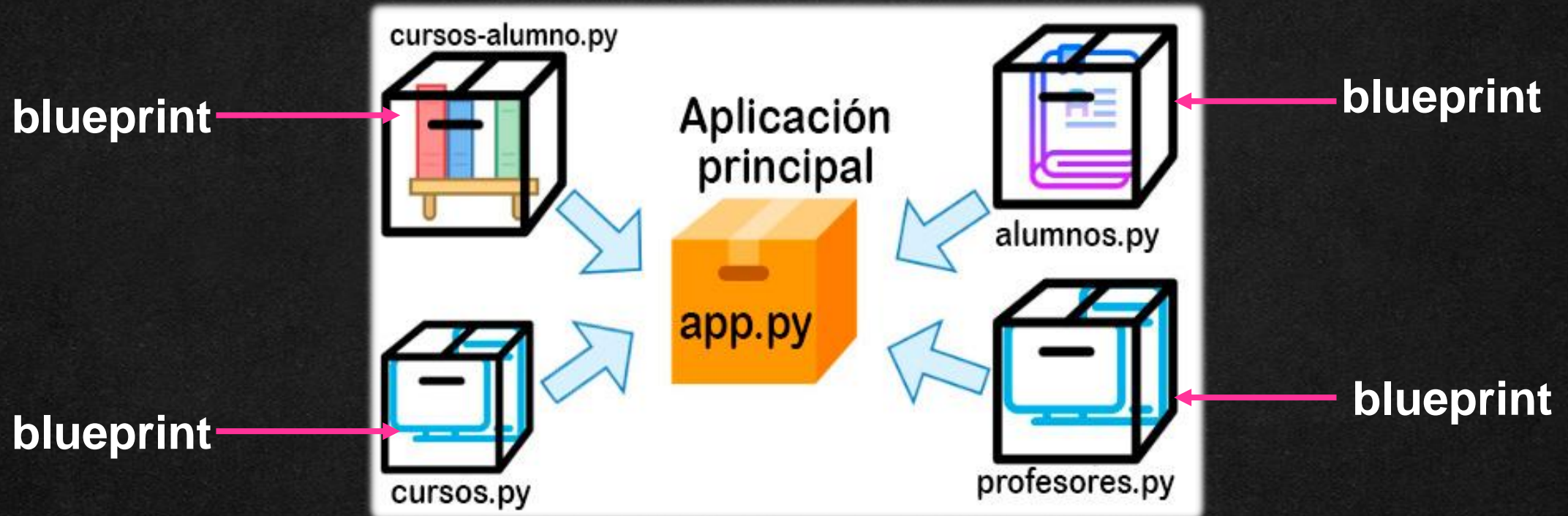




ProfesorenVideo.com

# Curso de Python

## Aplicación Modular







ProfesorenVideo.com

# Curso de Python

## Qué son los blueprint ?

Los **Blueprints** en Flask son una forma de estructurar y organizar una aplicación Flask en componentes reutilizables y modulares.

## Para qué sirven ?

- **Organización:** Facilitan la organización del código dividiendo la aplicación en módulos más manejables.
- **Reutilización:** Permiten reutilizar componentes en diferentes partes de la aplicación o incluso en diferentes aplicaciones.
- **Escalabilidad:** Hacen que la aplicación sea más fácil de escalar y mantener a medida que crece en tamaño y complejidad.
- **Colaboración:** Facilitan el trabajo en equipo, permitiendo que diferentes desarrolladores trabajen en diferentes módulos sin interferir entre sí.





ProfesorenVideo.com

# Curso de Python

Cómo estructurar la nueva aplicación para incorporar los blueprints ?

## ✓ MI-PROYECTO

> \_\_pycache\_\_

> blueprints

> models

> proyecto

> static

> templates

🔗 app.py

🔗 config.py

## ✓ blueprints

> \_\_pycache\_\_

🔗 alumnos\_blueprint.py

🔗 alumnos\_curso\_blueprint.py

🔗 cursos\_alumno\_blueprint.py

🔗 cursos\_blueprint.py

🔗 profesores\_blueprint.py



ProfesorenVideo.com

# Curso de Python

Si tengo una aplicación monolítica qué cosas debo cambiar en el código para hacerla modular ?

1.-) En la aplicación principal (**app.py**) :

- Debemos **registrar** los blueprints:

```
app.register_blueprint(alumnos_bp, url_prefix='/alumnos')
```

- Especificar la ruta **root** (raíz):

```
@app.route('/')  
def index():  
    return render_template('index.html')
```





ProfesorenVideo.com

# Curso de Python

Si tengo una aplicación monolítica qué cosas debo cambiar en el código para hacerla modular ?

2.-) Debemos crear un **blueprint** para agrupar la lógica de los procesos relacionados con una entidad fundamental (**ejemplo CRUD de tablas**). También para aquella lógica que involucra varias entidades del sistema.

```
✓ blueprints
  > __pycache__
  📁 alumnos_blueprint.py
  📁 alumnos_curso_blueprint.py
  📁 cursos_alumno_blueprint.py
  📁 cursos_blueprint.py
  📁 profesores_blueprint.py
```



ProfesorenVideo.com

# Curso de Python

Recordemos los nombres de funciones:

```
@alumnos_bp.route('/registrar', methods=['GET', 'POST'])  
def registra():  
    if request.method == 'POST':
```

```
@alumnos_bp.route('/consultar', methods=['GET', 'POST'])  
def consulta():  
    if request.method == 'POST':
```

```
@alumnos_bp.route('/eliminar/', methods=['GET', 'POST'])  
@alumnos_bp.route('/eliminar/<int:alumnoid>', methods=['GET', 'POST'])  
def elimina(alumnoid=None):  
  
    try:
```





ProfesorenVideo.com

# Curso de Python

Si tengo una aplicación monolítica qué cosas debo cambiar en el código para hacerla modular ?

3.-) Donde quiera que haya **un llamado** al nombre de una función, deberá agregarse lo siguiente:

`nombreDeBlueprint.nombreDeFunción( )`

```
li class="nav-item">
  <a class="nav-link" href="{{ url_for('alumnos.registra')}}">Registrar</a>
</li>
li class="nav-item">
  <a class="nav-link" href="{{ url_for('alumnos.consulta')}}">Consultar</a>
</li>
li class="nav-item">
  <a class="nav-link" href="{{ url_for('alumnos.modifica')}}">Modificar/Eliminar</a>
</li>
li class="nav-item">
  <a class="nav-link" href="{{ url_for('alumnos.lista')}}">Listar</a>
</li>
```





ProfesorenVideo.com

# Curso de Python

Si tengo una aplicación monolítica qué cosas debo cambiar en el código para hacerla modular ?

3.-) Donde quiera que haya una referencia al nombre de una función, deberá agregarse lo siguiente:

`nombreDeBlueprint.nombreDeFunción( )`

```
<tbody>
  {% for alumno in totalAlumnos %}
  <tr>
    <td>{{ alumno['AlumnoDNI'] }}</td>
    <td>{{ alumno['Apellido'] }}</td>
    <td>{{ alumno['Nombre'] }}</td>
    <td>{{ alumno['Email'] }}</td>
    <td>
      <a href="{{ url_for('alumnos.modifica', alumnoId=alumno['AlumnoID']) }}" class="btn btn-primary">Modificar</a>
      <a href="{{ url_for('alumnos.elimina', alumnoId=alumno['AlumnoID']) }}" class="btn btn-danger">Eliminar</a>
    </td>
  </tr>
  {% endfor %}
```



ProfesorenVideo.com

# Curso de Python

## DESPEDIDA

Estaré entregando TODA la información de sitios web de recursos que yo utilizo para mis desarrollos:

- Imágenes libres de derechos de autor
- Videos libres de derecho de autor
- Íconos gratis
- Libros gratis
- Y mucho más ...

Los espero en la despedida