



TECNOLÓGICO
NACIONAL DE MÉXICO




TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Cd. Altamirano
Subdirección Académica
Departamento de Sistemas y Computación

Academia de Informática

MANUAL DE PRÁCTICAS DE LA MATERIA
<< Programación en ambiente cliente servidor >>
IFF-1019


Elaborado por:
<<CARLOS ALBERTO BERNAL BELTRÁN>>

Periodo:
30 de enero al 02 de Junio 2023

	Manual de prácticas	Clave	IFF-1019
		Versión	2
	PROGRAMACIÓN EN AMBIENTE CLIENTE SERVIDOR	Pagina	Página 2 de 8
		Fecha de emisión	06/02/2023
Academia de Informática		Departamento de Sistemas y Computación	

INTRODUCCIÓN (DEL MANUAL DE PRÁCTICAS)

Competencia de la materia: Analiza y diseña soluciones de aplicaciones distribuidas utilizando los principales middlewares para el desarrollo de sistemas cliente/servidor.

	MANUAL DE PRACTICAS	Clave	IFF-1019
		Versión	1
	PROGRAMACIÓN EN AMBIENTE CLIENTE SERVIDOR	Página	Página 3 de 8
		Fecha de emisión	08/02/2022
Academia de Informática		Departamento de Sistemas y Computación	

PRÁCTICA No. 2

2.1 NOMBRE:

Desarrollar programas cliente-servidor utilizando Remote Method Invocation (RMI) como tecnología de base, e incluyendo serialización de objetos, control de políticas de seguridad y generación automática.

2.2 Competencia a desarrollar: Identificar las características, ventajas y desventajas del mecanismo RMI de Java para la intercomunicación de aplicaciones mediante la invocación de métodos remotos. Desarrollar aplicaciones empleando el mecanismo RMI.

2.3 Especificar la correlación con el o los temas y subtemas del programa de estudio vigente.

Tema 3:

RMI (REMOTE METHOD INVOCATION)

Subtemas:


- 3.1. Características y Estructura de RMI
- 3.2. El API Java RMI.
- 3.3. Jerarquía de objetos RMI.
- 3.4. El Sistema de Nombrado Registry.
- 3.5. Desarrollo de Aplicaciones Distribuidas.
- 3.6. Paso de parámetros a través de la red.
- 3.7. Callbacks (Resguardos).

2.4 Medidas de seguridad e higiene

Están establecidas dentro del manual de la sala de cómputo.

2.5 Materiales y equipo necesario

Computadoras de la sala de computo del Instituto Tecnológico de Cd. Altamirano.

	MANUAL DE PRACTICAS	Clave	IFF-1019
		Versión	1
	PROGRAMACIÓN EN AMBIENTE CLIENTE SERVIDOR	Página	Página 4 de 8
		Fecha de emisión	08/02/2022
Academia de Informática		Departamento de Sistemas y Computación	

Introducción

Pyro es un sistema de tecnología de **objetos distribuidos avanzado** y poderoso escrito completamente en Python, que está diseñado para ser muy fácil de usar. **Se asemeja a la invocación de método remoto (RMI) de Java.**

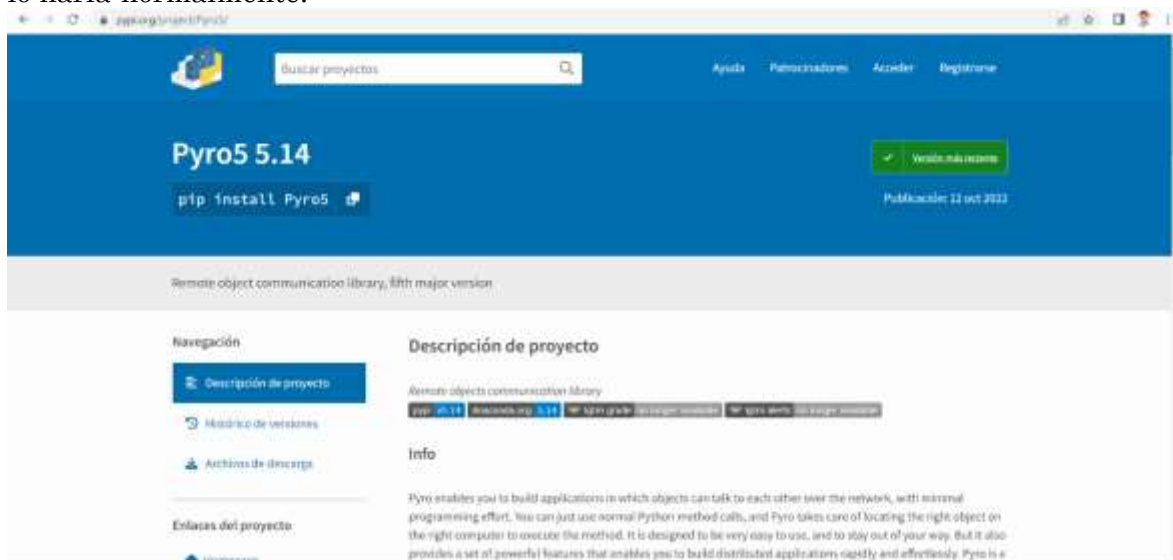
Pyro es una biblioteca que le **permite crear aplicaciones en las que los objetos pueden comunicarse entre sí a través de la red**, con un mínimo esfuerzo de programación. Simplemente puede usar llamadas de método Python normales, con casi todos los parámetros posibles y tipo de valor de retorno, y Pyro se encarga de ubicar el objeto correcto en la computadora correcta para ejecutar el método. Está diseñado para que sea muy fácil de usar y, por lo general, no estorbe. Pero también proporciona un conjunto de potentes funciones que le permiten crear aplicaciones distribuidas de forma rápida y sin esfuerzo. Pyro es una biblioteca de Python pura y se ejecuta en muchas plataformas y versiones de Python diferentes.

1.6. Metodología


En este manual indica que hacer para crear aplicaciones en las que los objetos pueden comunicarse entre sí a través de la red, con un mínimo esfuerzo de programación. Simplemente puede usar llamadas de métodos normales de Python para llamar a objetos en otras máquinas.

Pyro es una biblioteca de Python pura y se ejecuta en muchas plataformas y versiones de Python diferentes.

Pyro es la abreviatura de **PYthon Remote Objects**. Es un sistema avanzado y poderoso de **Tecnología de Objetos Distribuidos** escrito completamente en Python, que está diseñado para ser muy fácil de usar. No vuelva a preocuparse por escribir código de comunicación de red, cuando use Pyro simplemente **escriba sus objetos de Python** como lo haría normalmente.



Instalar Pyro5 en la PC

	MANUAL DE PRACTICAS	Clave	IFF-1019
		Versión	1
	PROGRAMACIÓN EN AMBIENTE CLIENTE SERVIDOR	Página	Página 5 de 8
		Fecha de emisión	08/02/2022
Academia de Informática		Departamento de Sistemas y Computación	

Para instalar pip en su sistema, siga estos pasos dependiendo de su sistema operativo: Windows

Descargue el script get-pip.py del sitio oficial: <https://bootstrap.pypa.io/get-pip.py>.

Abra la línea de comandos y vaya al directorio donde descargó el archivo get-pip.py.

Ejecute el siguiente comando: python get-pip.py.

Abra la terminal y escriba pip install Pyro5 para instalar la librería.

REALIZAR CAPTURAS DE LAS PRIMERAS PRUEBAS

Para programar con Pyro5, sigue estos pasos:

1. Instala Pyro5 en tu sistema usando pip. Abre una terminal o línea de comando y escribe lo siguiente:

```
pip install Pyro5
```

2. Importa Pyro5 en tu script de Python. Asegúrate de que la versión correcta de Pyro5 esté instalada. Puedes importar Pyro5 de la siguiente manera:

```
import Pyro5.api
```

3. Crea una clase que implemente los métodos que quieres exponer en la red. Por ejemplo:

```
class MyService:
    def hello ( self, name ):
        return "Hello, {}".format(name)
```

4. Crea un objeto de esta clase y lo expone a través de Pyro5. Para ello, crea un objeto de Pyro5 Daemon y registra el objeto en él:

```
from Pyro5.api import Daemon
daemon = Daemon()
service = MyService()
uri = daemon.register(service)
print(uri)
daemon.requestLoop()
```

En este ejemplo, uri es el identificador único que Pyro5 asigna al objeto registrado.

5. Ejecuta el daemon de Pyro5. En una terminal o línea de comando diferente, ejecuta el siguiente comando:


```
python <nombre_del_archivo>.py
```

6. Ahora, puedes invocar los métodos expuestos por el objeto registrado desde otros procesos que ejecuten Pyro5. Para ello, crea un proxy de Pyro5 para el objeto y llama a los métodos como si estuvieras trabajando con el objeto localmente:

```
import Pyro5.api
uri = "<el_identificador_uri_asignado_por_Pyro5>"
service = Pyro5.api.Proxy(uri)
print(service.hello("world"))
```

En este ejemplo, service es un proxy que apunta al objeto registrado previamente y se puede llamar a sus métodos de manera transparente.

7. Si no necesitas el objeto expuesto en la red, recuerda desregistrarlo para liberar los recursos que ocupa:

	MANUAL DE PRACTICAS	Clave	IFF-1019
		Versión	1
	PROGRAMACIÓN EN AMBIENTE CLIENTE SERVIDOR	Página	Página 6 de 8
		Fecha de emisión	08/02/2022
Academia de Informática		Departamento de Sistemas y Computación	

```
daemon.unregister(service)
```

Con estos pasos básicos, ya puedes programar con Pyro5 y exponer objetos de Python en la red. Consulta la documentación de Pyro5 para conocer todas las posibilidades que ofrece esta herramienta.

Pyro proporciona un *servidor de nombres*.

Nombre del servidor

Pyro Name Server es una herramienta que le ayuda a realizar un seguimiento de sus objetos en su red. también es una medida ns para dar a sus objetos Pyro nombres lógicos en lugar de la necesidad de saber siempre el nombre exacto del objeto (o id) y su ubicación.

Pyro nombrará sus objetos así:

PYRO:obj_dcf713ac20ce4fb2a6e72acaeba57dfd@localhost:51850

PYRO:custom_name@localhost:51851

Es una identificación de objeto única generada en un host determinado o un nombre que usted mismo eligió. Pero para conectarse a estos objetos, siempre necesitará saber el nombre o identificación exactos del objeto y el nombre de host y el número de puerto exactos del demonio Pyro donde se ejecuta el objeto. Esto puede volverse tedioso, y si mueve servidores (u objetos Pyro), sus programas cliente ya no podrán conectarse a ellos hasta que actualice todos los URI.


Ejecute en CMD: (ejemplo)

```
python -m Pyro5.nameserver
Not starting broadcast server for IPv6.
NS running on localhost:9090 (::1)
URI = PYRO:Pyro.NameServer@localhost:9090
```

En Pyro5, `nameserver.lookup` es un método que se utiliza para buscar un objeto registrado en el servidor de nombres Pyro. El método devuelve el URI del objeto si se encuentra en el servidor de nombres o levanta una excepción si el objeto no está registrado.

Aquí hay un ejemplo de cómo utilizar `nameserver.lookup` para buscar un objeto registrado en el servidor de nombres Pyro5:

```
import Pyro5.api
# crea el proxy del servidor de nombres Pyro5
nameserver = Pyro5.api.locate_ns()
# busca el objeto registrado con el nombre "objeto_personalizado"
uri = nameserver.lookup("objeto_personalizado")
# crea un proxy del objeto utilizando el URI devuelto por la búsqueda
objeto_proxy = Pyro5.api.Proxy(uri)
```

	MANUAL DE PRACTICAS		Clave	IFF-1019
			Versión	1
	PROGRAMACIÓN EN AMBIENTE CLIENTE SERVIDOR		Página	Página 7 de 8
			Fecha de emisión	08/02/2022
Academia de Informática		Departamento de Sistemas y Computación		

```
# llama a un método del objeto remoto utilizando el proxy
resultado = objeto_proxy.metodo_remoto(argumentos)
# imprime el resultado devuelto por el método remoto
print(resultado)
```

En este ejemplo, se crea un proxy para el servidor de nombres Pyro5 utilizando el método `locate_ns()`. Luego, se utiliza el método `lookup()` del proxy del servidor de nombres para buscar un objeto registrado con el nombre "objeto_personalizado".

Si se encuentra el objeto, se devuelve su URI y se crea un proxy del objeto utilizando ese URI. Finalmente, se llama a un método remoto del objeto utilizando el proxy y se imprime el resultado devuelto por el método remoto.

Es importante tener en cuenta que el nombre del objeto registrado debe ser único en el servidor de nombres Pyro5 para evitar conflictos con otros objetos registrados con el mismo nombre.


IMPORTANTE: El programa ahora necesita un servidor de nombres Pyro que se esté ejecutando. Puede iniciar uno escribiendo el siguiente comando: `python -m Pyro5.nameserver` en una ventana de consola separada (generalmente solo hay un servidor de nombres ejecutándose en su red). Después de eso, inicie el servidor y el cliente como antes.

```
C:\Windows\System32>python -m Pyro5.nameserver
Not starting broadcast server for IPv6.
NS running on localhost:9090 (:::1)
URI = PYRO:Pyro.NameServer@localhost:9090
```

En las clases de retroalimentación se esta agregando en las líneas de programación de Python, por ejemplo, como iniciar el nombre de servidor.

Pueden consultar todo el manual Pyro5 en <https://pyro5.readthedocs.io/en/stable/nameserver.html>

REALIZAR CAPTURAS DE LAS PRUEBAS Y DE LA TERMINAL COMPILADO CORRECTAMENTE.

	MANUAL DE PRACTICAS		Clave	IFF-1019
			Versión	1
	PROGRAMACIÓN EN AMBIENTE CLIENTE SERVIDOR		Página	Página 8 de 8
			Fecha de emisión	08/02/2022
Academia de Informática		Departamento de Sistemas y Computación		

1.7 Estructura del reporte del estudiante

EVIDENCIA DE APRENDIZAJE: A. El estudiante debe realizar un reporte, desarrollando correctamente el manual de prácticas sobre una aplicación que, mediante RMI, obtenga información del sistema de archivos de una PC remota y genere reportes, adaptándose a situaciones y contextos complejos (a) pudiendo trabajar en equipo al reflejar sus conocimiento de la realidad (a), donde integra los conocimientos del tema; propone y/o explica procedimientos no vistos en clase (c) o aprendidos en otra asignatura (b) haciendo aportaciones a las actividades académicas desarrolladas (b). Realiza su trabajo de manera autónoma y autorregulada (f).

PARA QUE LA EVIDENCIA PUEDA SER ENTREGADA, SE DEBERÁ MOSTRAR SU FUNCIONAMIENTO ANTE EL PROFESOR.

ARCHIVO EN PDF:

- Portada
- Índice
- Desarrollo
(Deberá presentar captura de pantalla y descripción de las pruebas de los ejercicios realizados en el código.)
- Referencias
- Conclusión