

**PATRONES DE DISEÑO CREACIONALES**

**MARTÍNEZ COSIO JOSÉ ALFREDO**

**Aarón Hernández García**

**Factory Method**

El **Factory Method** es un patrón de diseño creacional en programación que se usa para definir una interfaz o método abstracto para crear un objeto, pero dejando que las subclases decidan qué clase concreta instanciar.

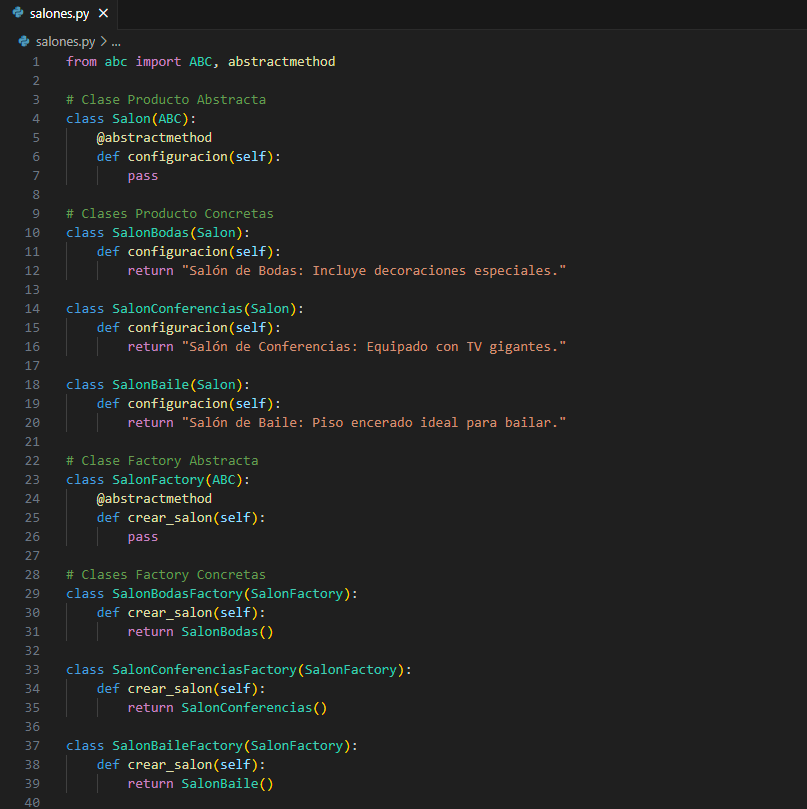
**09/11/2024**

**PÉREZ JUÁR HERNANDEZ GARCIA**

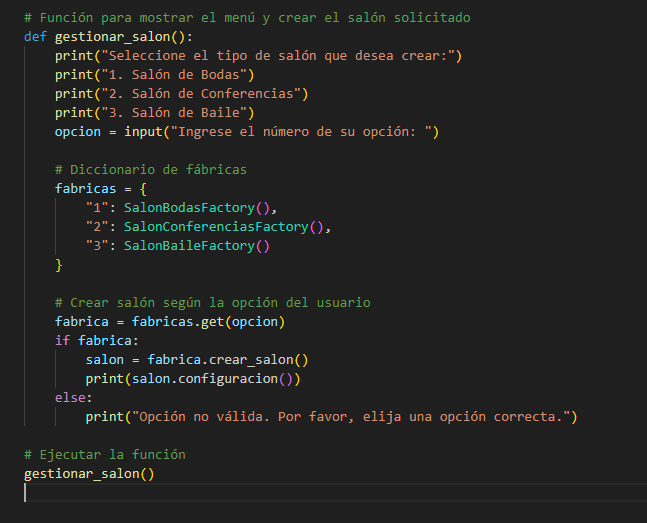
**1.2 APLICACIONES DE LA DISTRIBUCIÓN NORMAL**

**¿Por Qué Usar el Factory Method?**

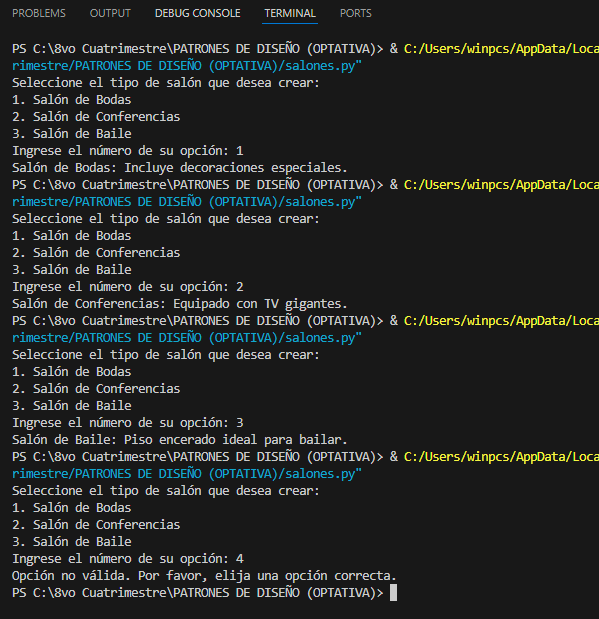
* **Desacoplamiento**: Permite crear objetos sin especificar la clase exacta de estos, lo cual hace que el código sea más flexible y fácil de mantener.
* **Extensibilidad**: Si necesitas agregar nuevos tipos de objetos, puedes hacerlo creando nuevas subclases en lugar de modificar el código existente.
* **Cambio Dinámico**: Facilita la creación de diferentes tipos de objetos en tiempo de ejecución, dependiendo de la situación o los parámetros.

**Código en Python ->**

**Código en Python ->**



**Respuesta ->**



**Referencias ->**

*Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.*

*Freeman, E., & Robson, E. (2004). Head First Design Patterns. O'Reilly Media.*

*Refactoring Guru. (n.d.). Factory Method. Recuperado de* [*https://refactoring.guru/design-patterns/factory-method*](https://refactoring.guru/design-patterns/factory-method)*.*