

## 1. Diagrama de Casos de Uso para el Sistema de Reservas de Espacios de Coworking.

### Actores:

- **Usuario de Coworking:** Cualquier persona que desee reservar un espacio de coworking.
- **Administrador de Coworking:** Persona responsable de gestionar el sistema de reservas, incluyendo el registro de usuarios, la adición de espacios y la gestión general de reservas.

### Casos de Uso Principales:

#### Para el Usuario de Coworking:

- **Crear Reserva:** El usuario selecciona un espacio disponible y realiza una reserva para una fecha y hora específicas.
- **Cancelar Reserva:** El usuario cancela una reserva existente.
- **Ver Espacios Disponibles:** El usuario consulta los espacios de coworking disponibles para reservar.

#### Para el Administrador de Coworking:

- **Registrar Usuario:** El administrador añade un nuevo usuario al sistema, permitiéndole hacer reservas.
- **Añadir Espacio:** El administrador registra un nuevo espacio de coworking en el sistema, incluyendo detalles como la capacidad y disponibilidad.
- **Gestionar Reservas:** El administrador revisa, modifica o cancela reservas existentes.
- **Actualizar Espacio:** El administrador actualiza la información o la disponibilidad de un espacio existente.

### Relaciones:

- Cada caso de uso se conecta al actor correspondiente mediante una línea, indicando quién puede iniciar cada acción dentro del sistema.
- Los casos de uso como "Crear Reserva" y "Cancelar Reserva" estarán directamente relacionados con el "Usuario de Coworking".
- Los casos de uso de gestión ("Registrar Usuario", "Añadir Espacio", "Gestionar Reservas", "Actualizar Espacio") estarán conectados al "Administrador de Coworking".

## **Relación entre crear reserva y ver espacios disponibles**

### **1. Dependencia:**

- "Crear Reserva" depende de "Ver Espacios Disponibles". Esto significa que el usuario debe ver los espacios disponibles antes de poder crear una reserva.
- Se puede representar mediante una flecha que va desde "Ver Espacios Disponibles" hacia "Crear Reserva".

### **2. Inclusión:**

- "Ver Espacios Disponibles" puede ser un paso dentro del caso de uso "Crear Reserva". Esto significa que el proceso de crear una reserva incluye la posibilidad de ver los espacios disponibles.
- Se puede representar mediante una línea punteada que conecta "Ver Espacios Disponibles" con "Crear Reserva", y una etiqueta que indique la relación, como "incluye" o "comprende".

La elección de la forma de representar la relación depende del nivel de detalle que se quiera mostrar en el diagrama. La dependencia es más simple, mientras que la inclusión es más específica y describe mejor el flujo de trabajo.

## **Relación entre registrar usuario y gestionar reservas**

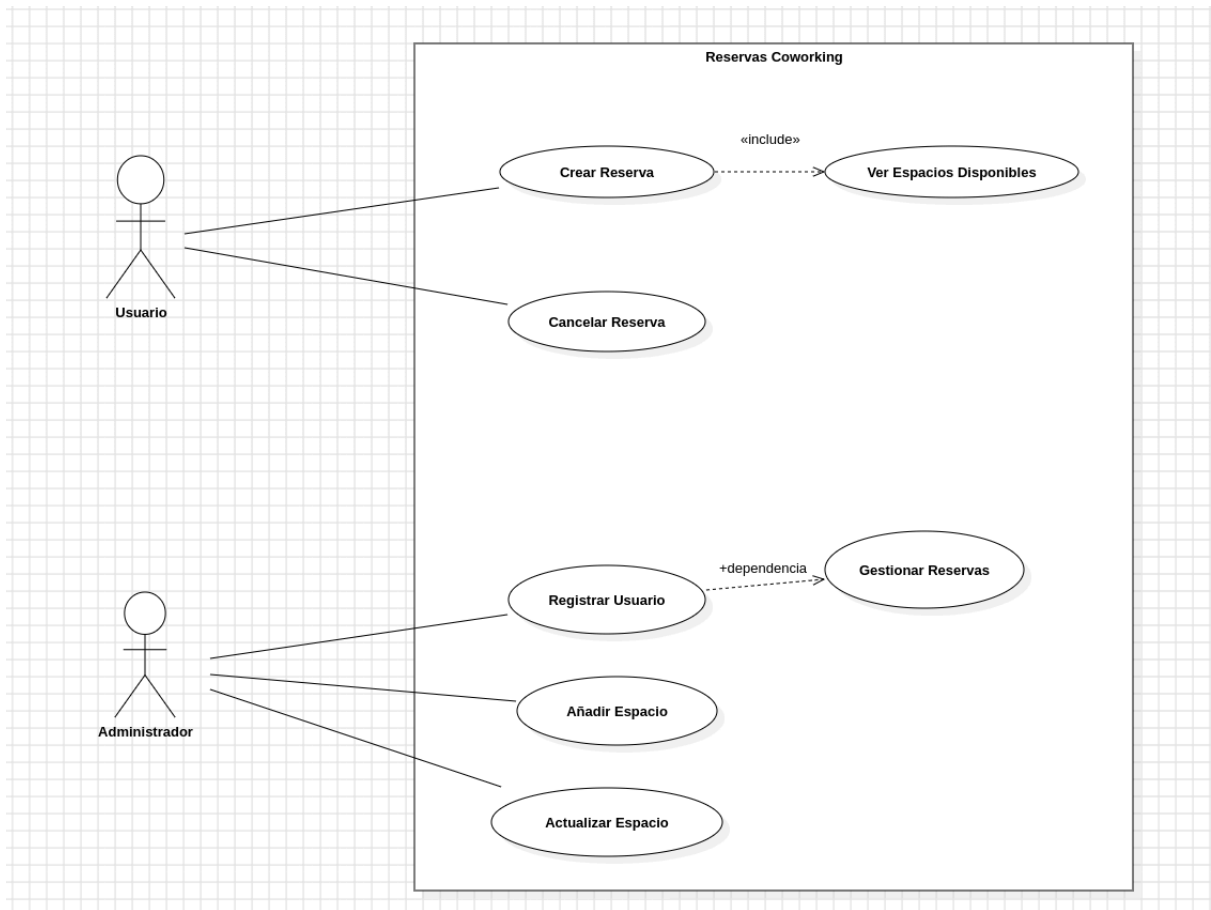
### **1. Dependencia:**

- "Gestionar Reservas" depende de "Registrar Usuario". Esto significa que solo los usuarios registrados pueden realizar reservas.
- Se puede representar mediante una flecha que va desde "Registrar Usuario" hacia "Gestionar Reservas".

### **2. Inclusión:**

- "Registrar Usuario" puede ser un paso dentro del caso de uso "Gestionar Reservas". Esto significa que el proceso de gestionar reservas puede incluir la posibilidad de registrar un nuevo usuario.
- Se puede representar mediante una línea punteada que conecta "Registrar Usuario" con "Gestionar Reservas", y una etiqueta que indique la relación, como "incluye" o "comprende".

La elección de la forma de representar la relación depende del nivel de detalle que se quiera mostrar en el diagrama. La dependencia es más simple, mientras que la inclusión es más específica y describe mejor el flujo de trabajo.



## 2. Diagrama de Clases para el Sistema de Reservas de Espacios de Coworking

### Clases Principales y Sus Relaciones:

1. **Clase Usuario:**
  - **Atributos:** nombre (str), email (str), telefono (str).
  - **Métodos:** mostrar\_info().
  - Relación: Tiene 0 o muchas reservas (Reserva).
2. **Clase Espacio:**
  - **Atributos:** identificador (str), tipo (str), capacidad (int), disponibilidad (bool).
  - **Métodos:** reservar(), cancelar\_reserva(), verificar\_disponibilidad().
  - Relación: Participa en 0 o muchas reservas (Reserva).
3. **Clase Reserva:**
  - **Atributos:** usuario (Usuario), espacio (Espacio), fecha (datetime), hora (datetime), duracion (int).
  - **Métodos:** confirmar\_reserva(), cancelar\_reserva().
  - Relación: Asocia 1 Usuario con 1 Espacio.
4. **Clase GestorReservas:**
  - **Atributos:** usuarios (dict), espacios (dict), reservas (list).
  - **Métodos:** registrar\_usuario(), añadir\_espacio(), crear\_reserva(), cancelar\_reserva().
  - Relación: Contiene y gestiona instancias de Usuario, Espacio, y Reserva.

### Representación de las Relaciones:

- **Asociación:** Usa líneas sólidas para conectar **Reserva** con **Usuario** y **Espacio**, mostrando la asociación directa.
- **Agregación o Composición:** Representa la relación entre **GestorReservas** y las otras clases (Usuario, Espacio, Reserva) mediante líneas con un diamante en el extremo de **GestorReservas**, indicando que **GestorReservas** contiene y gestiona estas entidades.

