

```
In [1]: class Accion:
    def __init__(self, nombre='', aplicabilidad=None, aplicacion=None,
                 coste=None):
        self.nombre = nombre
        self.aplicabilidad = aplicabilidad
        self.aplicacion = aplicacion
        self.coste = coste

    def es_aplicable(self, estado):
        if self.aplicabilidad is None:
            raise NotImplementedError(
                'Aplicabilidad de la acción no implementada')
        else:
            return self.aplicabilidad(estado)

    def aplicar(self, estado):
        if self.aplicar is None:
            raise NotImplementedError(
                'Aplicación de la acción no implementada')
        else:
            return self.aplicacion(estado)

    def coste_de_aplicar(self, estado):
        if self.coste is None:
            return 1
        else:
            return self.coste(estado)

    def __str__(self):
        return 'Acción: {}'.format(self.nombre)

class ProblemaEspacioEstados:
    def __init__(self, acciones, estado_inicial=None, estados_finales=None):
        if not isinstance(acciones, list):
            raise TypeError('Debe proporcionarse una lista de acciones')
        self.acciones = acciones
        self.estado_inicial = estado_inicial
        self.estados_finales = estados_finales

    def es_estado_final(self, estado):
        return estado in self.estados_finales

    def acciones_aplicables(self, estado):
        return (accion
                for accion in self.acciones
                if accion.es_aplicable(estado))
```