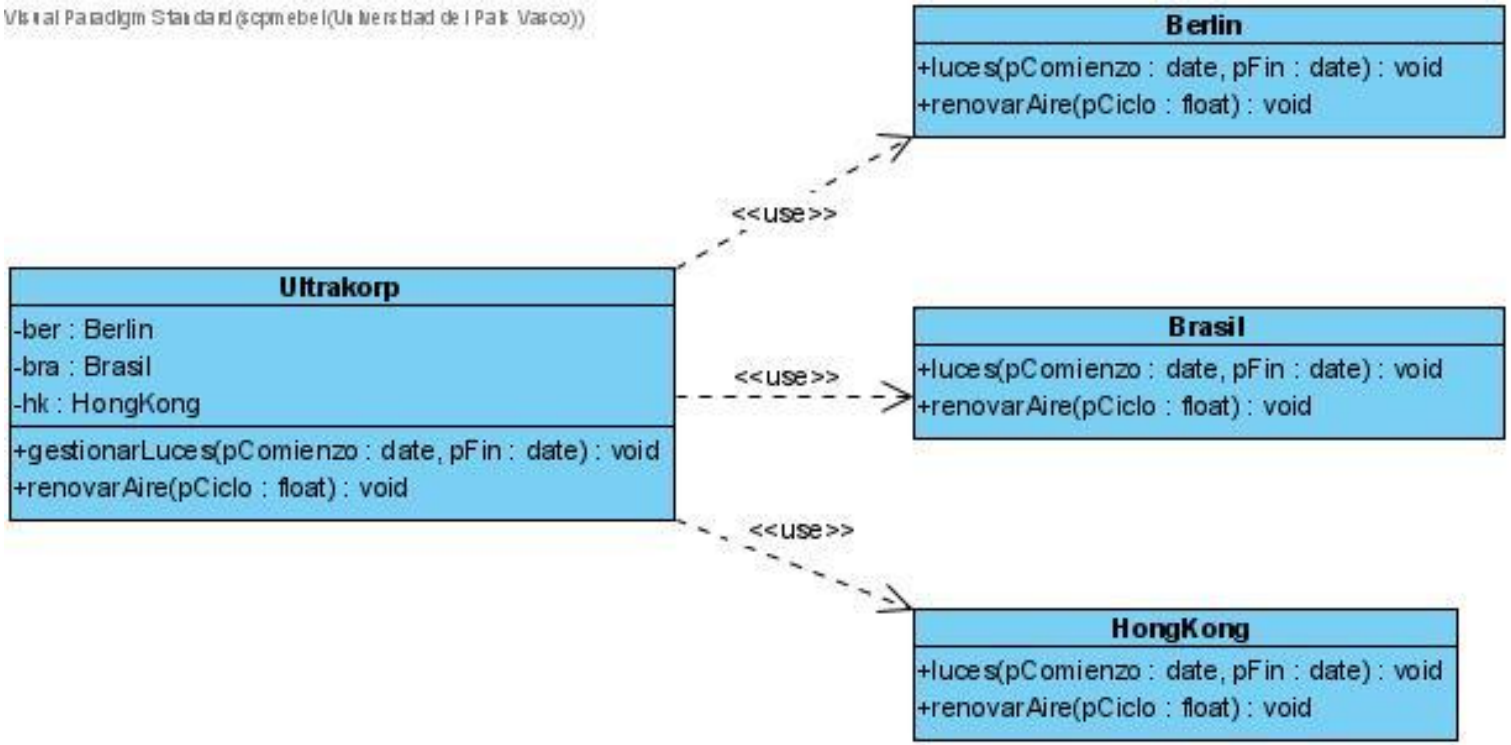
DIAGRAMA DE CLASES (SOLUCIÓN FACADE) (HAY DOS SOLUCIONES)

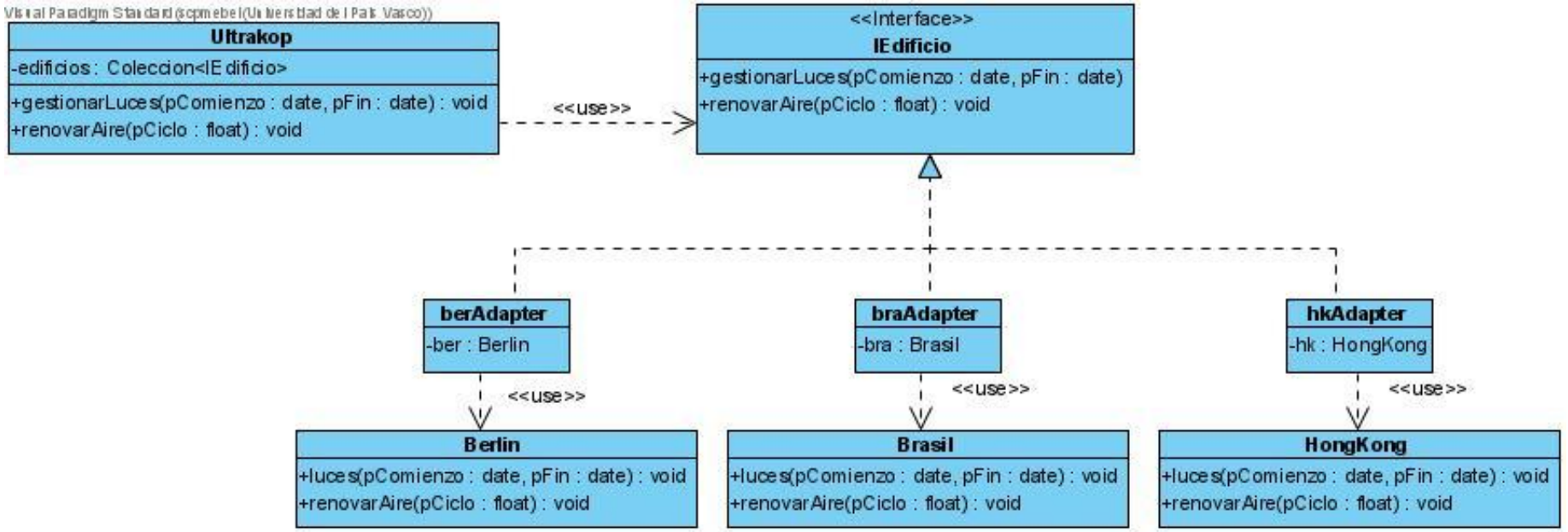


CÓDIGO (SOLUCIÓN FACADE)

```
public class Ultrakorp {

	private Berlin ber;
	private Brasil bra;
	private HongKong hk;

	public void gestionarLuces(Date pComienzo, Date
pFin) {
		ber.luces(pComienzo, pFin);
		bra.luces(pComienzo, pFin);
		hk.luces(pComienzo, pFin);
	}

	public void renovarAire(float pCiclo) {
		ber.renovarAire(pCiclo);
		bra.renovarAire(pCiclo);
		hk.renovarAire(pCiclo);
	}
}
```

```
public class Berlin {
public void luces(Date pComienzo, Date pFin) {
		System.out.println("Luces Berlin");
	}
	public void renovarAire(float pCiclo) {
		System.out.println("Renovar aire Berlin");
	}
}

public class Brasil {
	public void luces(Date pComienzo, Date pFin) {
		System.out.println("Luces Brasil");
	}
	public void renovarAire(float pCiclo) {
		System.out.println("Renovar aire Brasil");
	}
}

public class HongKong {
	public void luces(Date pComienzo, Date pFin) {
		System.out.println("Luces HongKong");
	}
	public void renovarAire(float pCiclo) {
		System.out.println("Renovar aire
HongKong");
	}
}
```

DIAGRAMA DE CLASES (SOLUCIÓN ADAPTER)



CÓDIGO (SOLUCIÓN ADAPTER)

```java
public class Ultrakop {
	private List<IEdificio> edificios = new
ArrayList<IEdificio>();
	public void gestionarLuces(Date pComienzo, Date pFin) {
		Iterator<IEdificio> it = edificios.iterator();
		while(it.hasNext())
		{
			it.next().gestionarLuces(pComienzo, pFin);
		}
	}
	public void renovarAire(float pCiclo) {
		Iterator<IEdificio> it = edificios.iterator();
		while(it.hasNext())
		{
			it.next().renovarAire(pCiclo);
		}
	}
}
```

```java
public interface IEdificio {
	void gestionarLuces(Date pComienzo, Date pFin);
	void renovarAire(float pCiclo);
}
```

```java
public class berAdapter implements IEdificio {
	private Berlin ber;
	@Override
	public void gestionarLuces(Date pComienzo, Date pFin) {
		ber.luces(pComienzo, pFin);
	}
	@Override
	public void renovarAire(float pCiclo) {
		ber.renovarAire(pCiclo);
	}
}
```

```java
public class braAdapter implements IEdificio {
	private Brasil bra;
	@Override
	public void gestionarLuces(Date pComienzo, Date pFin) {
		bra.luces(pComienzo, pFin);
	}
	@Override
	public void renovarAire(float pCiclo) {
		bra.renovarAire(pCiclo);
	}
}
```

```java
public class hkAdapter implements IEdificio {
	private HongKong hk;
	@Override
	public void gestionarLuces(Date pComienzo, Date pFin) {
		hk.luces(pComienzo, pFin);
	}
	@Override
	public void renovarAire(float pCiclo) {
		hk.renovarAire(pCiclo);
	}
}
```