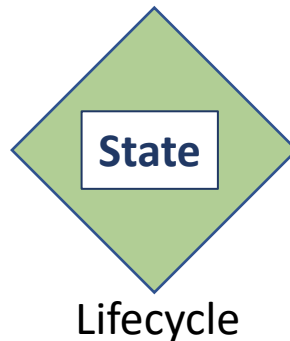


LiveCycle

Lifecycle

Es un *holder* que almacena información referente al estado del ciclo de vida (*lifecycle*) de un componente. Este componente, por ejemplo AppCompatActivity, debe ser un **LifecycleOwner**,.

```
Lifecycle.State state = lifecycle.getCurrentState()
```

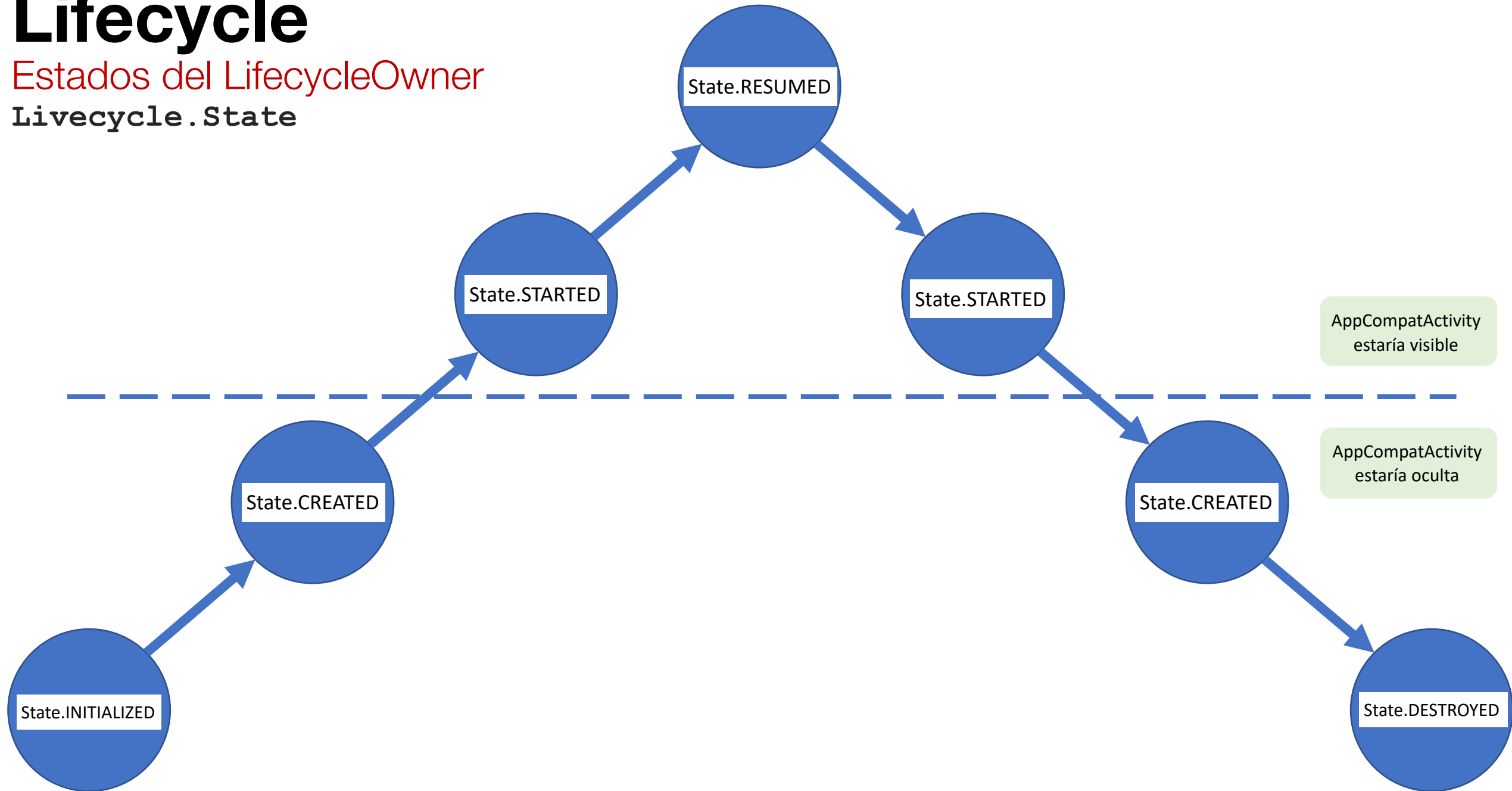


```
public interface LifecycleOwner
{
    Lifecycle getLifecycle();
}
```

Lifecycle

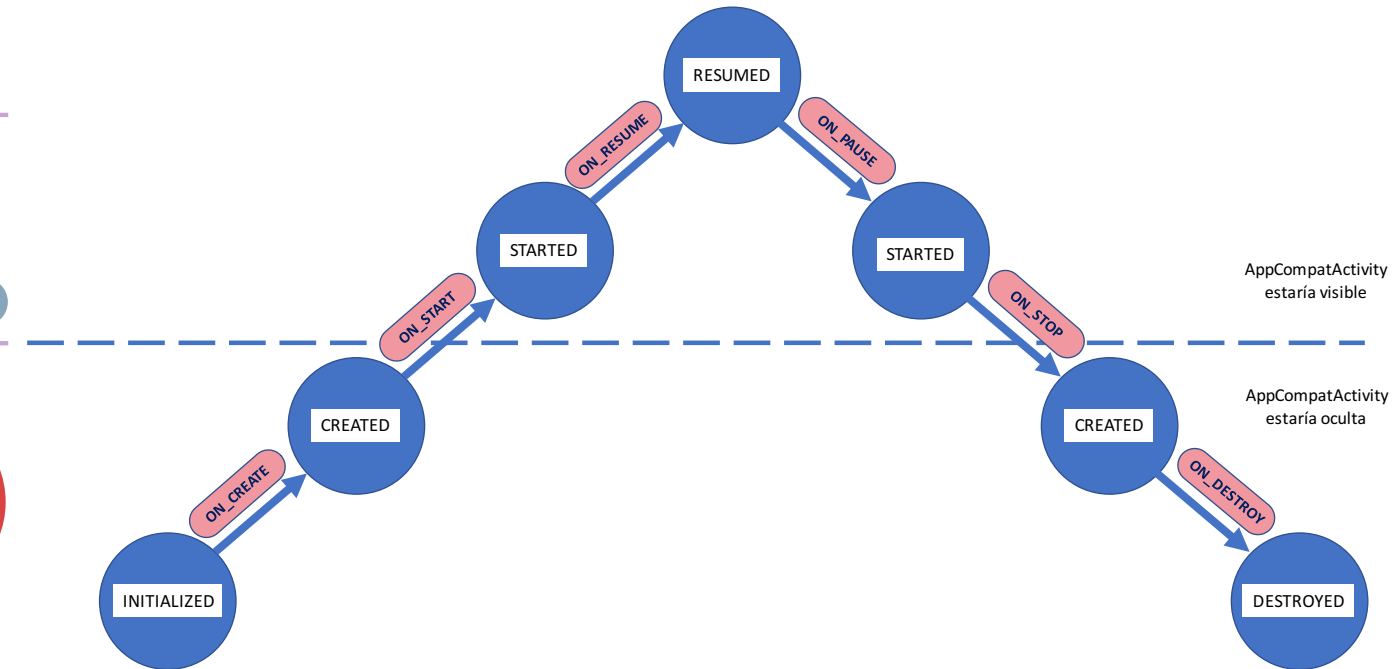
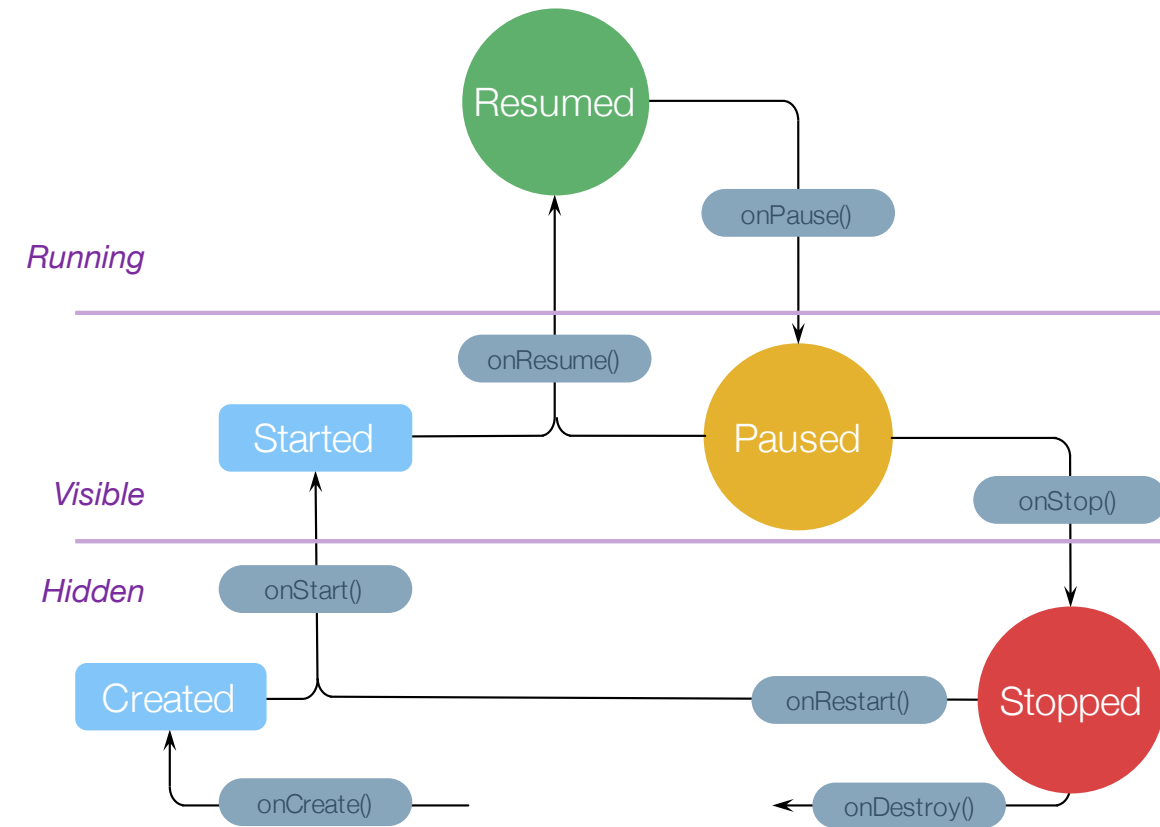
Estados del LifecycleOwner

Lifecycle.State



Lifecycle

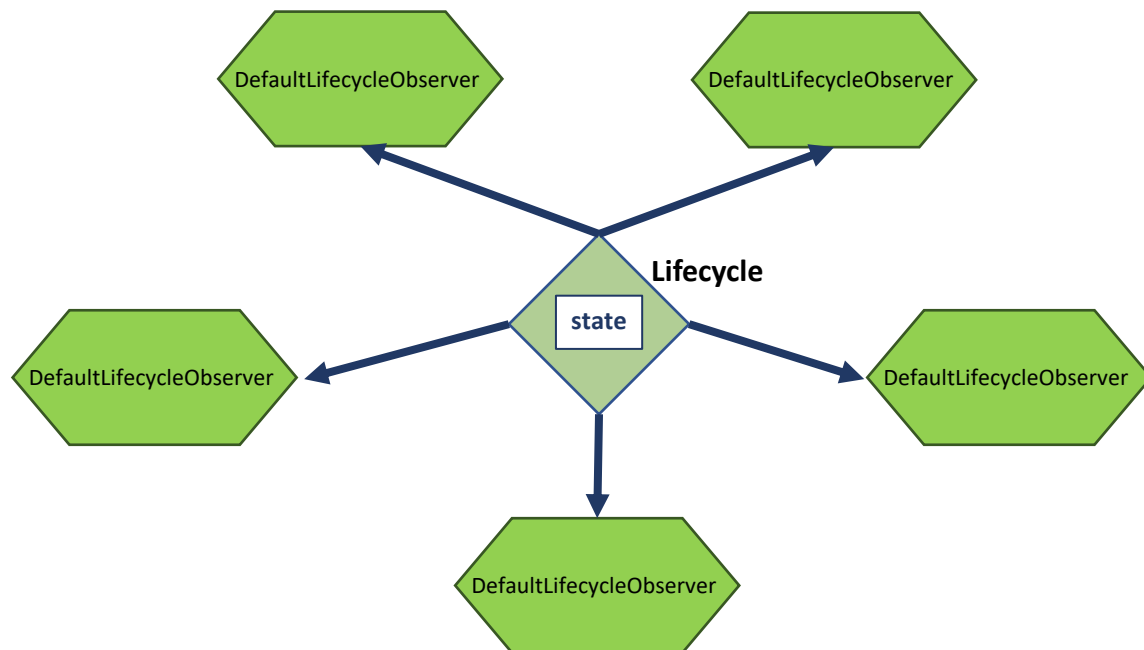
Estados de Activity vs. LifecycleOwner



Lifecycle

Observable

Lifecycle es un *observable* y como tal permite que *observers* (instancias **LifecycleObserver**) puedan suscribirse y recibir notificaciones cuando se cambie el estado del ciclo de vida de su LifecycleOwner.



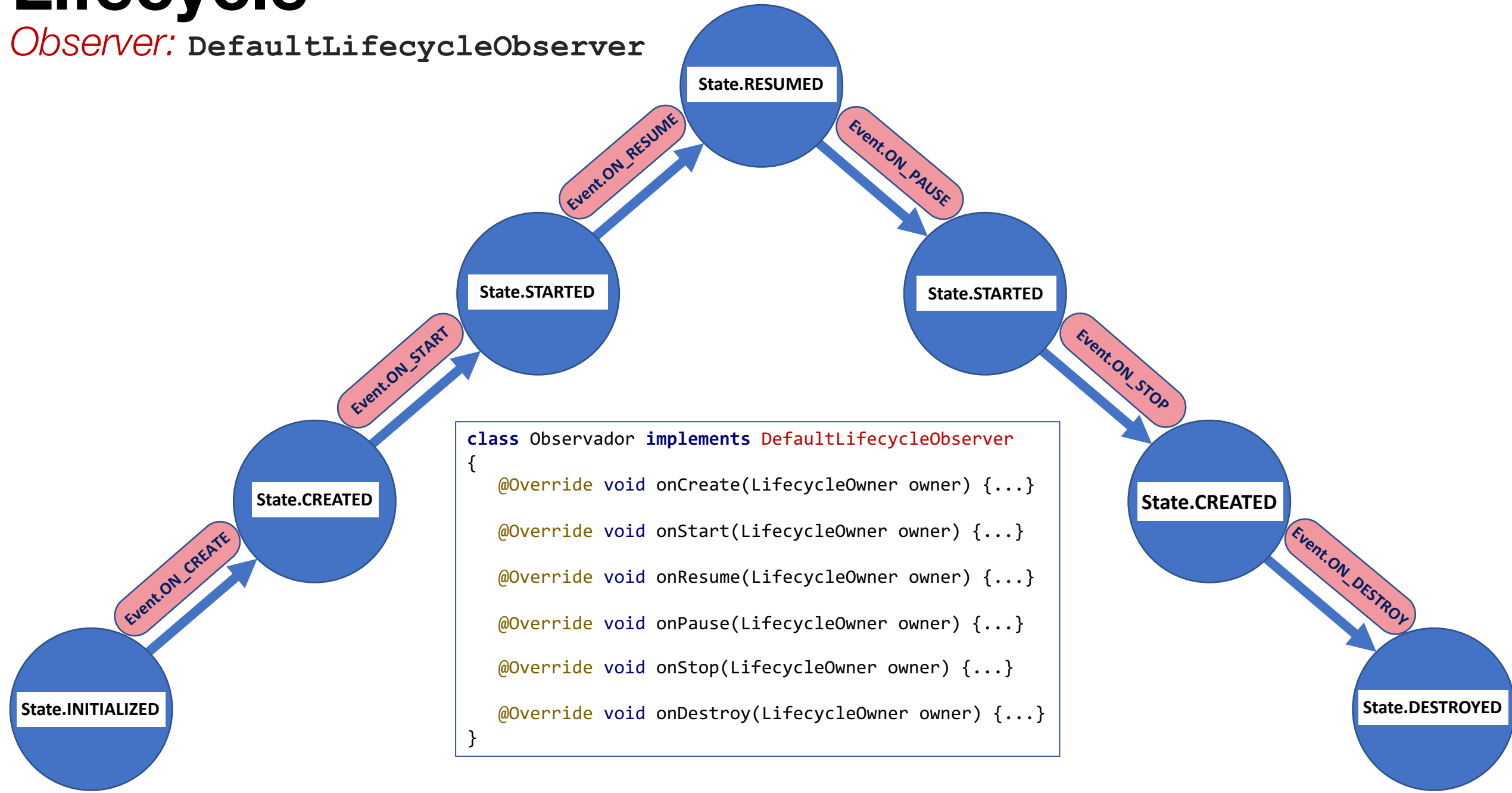
En Java8, los *observers* serán instancias de **DefaultLifecycleObserver**

```
public class Observador implements DefaultLifecycleObserver
{
    : : :
}

lifecycle.addObserver( new Observador() );
```

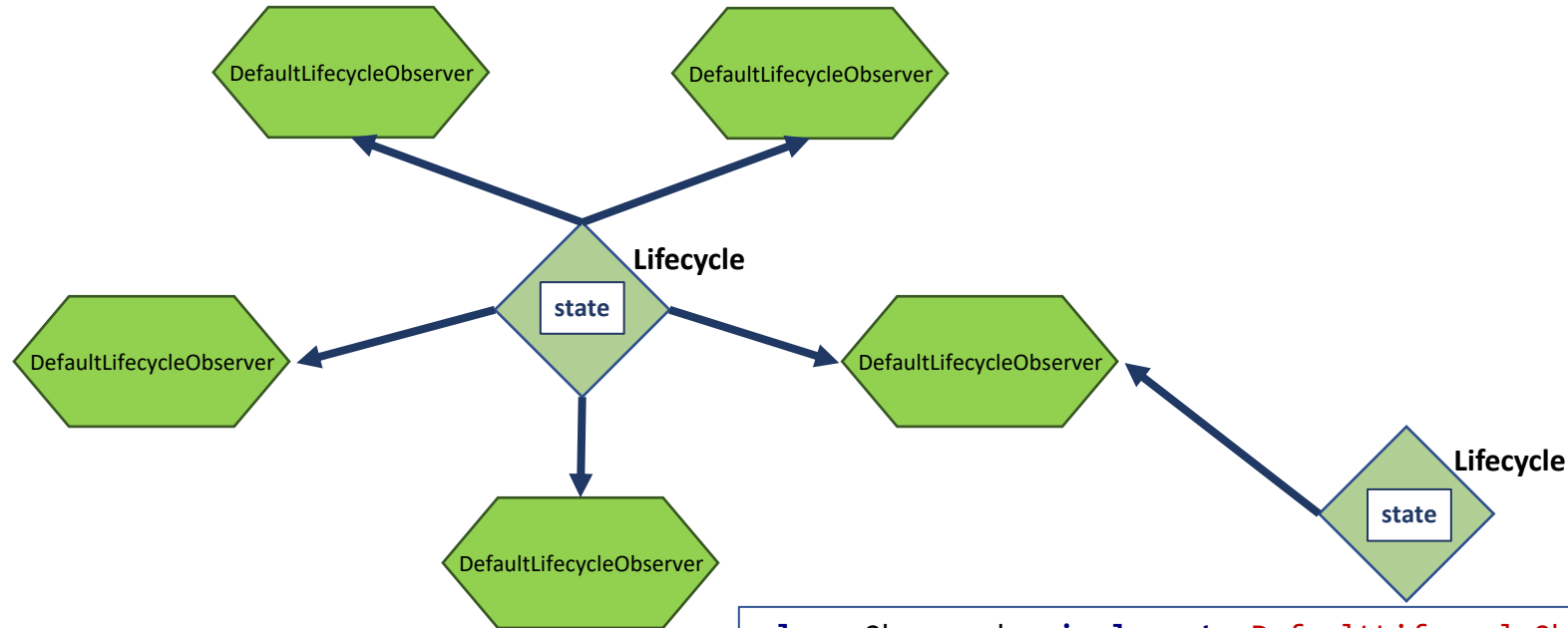
Lifecycle

Observer: DefaultLifecycleObserver



Lifecycle

Observer: DefaultLifecycleObserver



```
class Observador implements DefaultLifecycleObserver
{
    @Override void onCreate(LifecycleOwner owner) {...}
    @Override void onStart(LifecycleOwner owner) {...}
    @Override void onResume(LifecycleOwner owner) {...}
    @Override void onPause(LifecycleOwner owner) {...}
    @Override void onStop(LifecycleOwner owner) {...}
    @Override void onDestroy(LifecycleOwner owner) {...}
}
```

Lifecycle

Observer: LifecycleObserver

Lifecycle.Event + LifecycleObserver

