

## Contratos de Operaciones

### Índice de contenido

Contratos de operaciones.....	1
Contratos de operaciones del sistema.....	1
Ejemplo de contrato.....	1
Secciones de un Contrato.....	2
Construcción de un Contrato.....	2
Contratos en UML.....	2
Contratos en el UP.....	2
Referencias y lecturas recomendadas.....	3
Lecturas recomendadas.....	3
Referencias.....	3

### Contratos de operaciones del sistema.

Los contratos de operaciones ayudan a describir el comportamiento del sistema. Un *contrato de operación* describe el cambio de un estado a otro de los objetos del dominio como resultado de la ejecución de una operación. Su creación depende de la creación previa del modelo de dominio, los diagramas de secuencia de sistema, y la identificación de las operaciones de sistema. Los contratos de operaciones se refieren a las *operaciones del sistema*, las operaciones ofrecidas por el sistema como caja negra a través de su interfaz pública de datos en respuesta a los eventos entrantes al sistema. El sistema como un todo se puede representar en UML como una clase.

Los casos de uso constituyen el principal mecanismo de descubrimiento del comportamiento del sistema; los contratos de operación no son obligatorios, se construyen cuando los detalles y complejidad de los cambios de estado implicados en la operación hacen difícil su captura en los casos de uso.

Los contratos suelen escribirse en forma declarativa, indicando las condiciones previas y posteriores a la operación.

### Ejemplo de contrato.

El siguiente ejemplo de contrato corresponde a la operación `ingresarArticulo(idArticulo, cantidad)` identificada en el diagrama de secuencia del sistema. UPC, Universal Product Code, es el código universal de producto, expresable como un número o en código de barras. La etiqueta de los productos muestran el UPC tanto en código de barras como en forma numérica legible por humanos.

**Nombre:** `ingresarArticulo` (`idArticulo`: UPC, `cantidad`: entero)

**Referencias cruzadas:** `RegistrarVenta` (caso de uso)

**Precondiciones:** Venta en curso.

**Postcondiciones:**

- se creó `idv`, instancia de `LineaDeVenta` (creación de instancia).
- `idv` se asoció con la Venta actual (formación de asociación)
- `idv.cantidad` pasó a ser `cantidad` (modificación de atributo)

- idv se asoció con una EspecificacionDeProducto en base a coincidencia de idArticulo (formación de asociación).

## Secciones de un Contrato.

**Nombre:** nombre de la operación y parámetros.

**Referencias cruzadas:** casos de uso en los que puede realizarse la operación.

**Precondiciones:** estado del sistema antes de la ejecución. Suposiciones relevantes sobre el estado del sistema o de los objetos del modelo de dominio antes de la ejecución de la operación. La operación no verifica estas condiciones, las da por cumplidas. En el flujo de ejecución del programa el analista deberá asegurar la consecución de las condiciones previas por la ejecución exitosa de operaciones anteriores y el cumplimiento de las restricciones requeridas.

**Postcondiciones:** estado del sistema después de la ejecución de esta operación. El estado del sistema está dado por el estado de los objetos del modelo de dominio después de completada la operación. Comprende la creación de instancias, la formación o rotura de asociaciones, y los cambios en los atributos. A nivel conceptual la eliminación explícita de instancias no suele tener relevancia; la eliminación de objetos es más un problema de implementación, por la liberación de recursos, algo ajeno al modelo conceptual en construcción en estas etapas. La expresión de las postcondiciones es en tiempo pasado: se hicieron tales y tales cosas.

## Construcción de un Contrato.

Para cada caso de uso:

1. Identificar las operaciones del sistema a partir de los diagramas de secuencia del sistema.
2. Construir contratos para las operaciones del sistema que resulten complejas, sutiles en sus resultados, o poco claras en la descripción del caso de uso.
3. Escribir las precondiciones.
4. Escribir las postcondiciones, usando estas categorías:
  - creación y eliminación de instancias.
  - modificación de atributos.
  - formación y destrucción de asociaciones.
5. Revisar las precondiciones, si están todas las necesarias y sólo las necesarias.

Las precondiciones son una foto antes de la operación; las postcondiciones son una foto después de la operación.

## Contratos en UML.

En UML los contratos son especificaciones de operaciones. Formalmente, en UML,

- una *operación* es una especificación de una transformación o consulta que se puede invocar para que la ejecute un objeto.
- un *método* es la implementación de una operación.
- una operación tiene una *signatura* (nombre y lista de parámetros), un *valor de retorno* y una *especificación de operación* descriptiva de los efectos producidos por la ejecución de esa operación (las postcondiciones).

En UML las operaciones pertenecen a las clases. En UML los subsistemas se modelan como clases o como paquetes. El sistema global se modela como una clase habitualmente denominada Sistema con operaciones públicas acompañadas de sus especificaciones de operación correspondientes.

## Contratos en el UP.

Los contratos, si se usan, se escriben en la fase de Elaboración. Escribir contratos sólo para las operaciones del sistema más complejas y sutiles.

## Referencias y lecturas recomendadas.

El contenido de este documento está basado en las fuentes citadas a continuación, cuya lectura o consulta no pretenden sustituir.

### Lecturas recomendadas.

- [Larman2003] Larman, Craig. UML y patrones. Una introducción al análisis y diseño orientado a objetos y al Proceso Unificado, 2a. edición. Madrid, 2003. ISBN 84-205-3438-2.
- [Fowler1997] Fowler, Martin y Scott, Kendall. UML distilled. Applying the Standard Object Modelling Language. Addison Wesley Longman, Inc., 1997. ISBN 0-201-32563-2.
- [Pfleeger2002] PFLEEGER, SHARI LAWRENCE. Ingeniería de software, teoría y práctica, 1a. edición. Buenos Aires, Pearson educación, 2002. ISBN: 987-9460-71-5.

### Referencias.

- [Jacobson2000] Jacobson, Ivar; Booch, Grady; Rumbaugh, James. El proceso unificado de desarrollo de software. Pearson Educación, Madrid, 2000. ISBN: 84-7829-036-2.
- [Booch1999] Booch, Grady; Jacobson, Ivar; Rumbaugh, James. El lenguaje unificado de modelado. Pearson Educación, Madrid, 2000. ISBN: 84-7829-028-1.

Errores, omisiones, comentarios: Víctor González Barbone, [vagonbar en fing.edu.uy](mailto:vagonbar@fing.edu.uy)

---

*Desarrollo de Software para Ingeniería Eléctrica - Rev. 2009-05-09*

[Instituto de Ingeniería Eléctrica - Facultad de Ingeniería - Universidad de la República, Uruguay.](#)