

## AUTOEVALUACIÓN 1 → Conceptos de Ingeniería de Software y Técnicas de Elicitación

1. Según Richard Stallman el término software libre (o programas libres) se refiere a libertad. En concreto se refiere a cuatro libertades:

- De ejecutar el programa en cualquier sitio, con cualquier propósito y para siempre.
- De estudiarlo y adaptarlo a nuestras necesidades. Esto exige el acceso al código fuente.
- De redistribución, de modo que se nos permita colaborar con vecinos y amigos.
- De no cobrar nunca por el desarrollo ni poder venderlo.

**FALSO:** De no cobrar nunca por el desarrollo ni poder venderlo. → Libertad de mejorar y publicar las mejoras del programa. Exige el acceso a código fuente.

2. Complete el texto con las palabras correctas:

El entrevistador debe vestirse adecuadamente y ser cortés durante toda la entrevista. Una vez iniciada la entrevista debe escuchar cuidadosamente y debe evitar revelar pistas. Debe observar los gestos y evitar hablar en lugar de escuchar. El entrevistador debe ser paciente y evitar el uso de grabadores.

3. La observación del ambiente de trabajo es una técnica de elicitación de recolección de hechos a partir de la documentación existente.

**FALSO:** La observación del ambiente de trabajo es una técnica de elicitación en donde el analista se convierte en observador de las personas y actividades con el objetivo de aprender acerca del sistema.

4. Los usuarios finales, ingenieros, gerentes y expertos del dominio son stakeholders. **VERDADERO.**

5. Seleccione de las siguientes técnicas de elicitación aquellas que se corresponden con **métodos discretos** de recolección de información.

- Brainstorming. → **Método interactivo.**
- **Muestreo de la documentación.**
- **Observación del ambiente de trabajo.**
- **Investigación y visitas al lugar.**
- Cuestionarios. → **Método interactivo.**
- Entrevistas. → **Método interactivo.**

6. Cuando hablamos de “requerimientos del cliente” estamos haciendo referencia a las “necesidades del cliente”. **VERDADERO.**

7. El problema del envejecimiento del Software NO está en la operación, sino en los cambios que se introducen. **VERDADERO.**

8. El punto de vista del “interactuador” representa las características y restricciones del dominio que influyen en los requerimientos del sistema.

**FALSO:** El punto de vista del interactuador representa a las personas u otros sistemas que interactúan directamente con el sistema, y que pueden influir en los requerimientos del sistema de algún modo.

9. Seleccione aquellas afirmaciones que se corresponden con el tipo de preguntas que se arman en un **cuestionario**:

- La facilidad de análisis de una pregunta abierta es fácil. → **Es difícil.**
- La preparación de una pregunta cerrada es fácil. → **Es difícil.**
- **La pregunta abierta tiene una velocidad de conclusión lenta.**
- La pregunta cerrada tiene una naturaleza exploratoria alta. → **Baja.**
- **La amplitud y profundidad en una pregunta abierta es alta.**

10. Seleccione las afirmaciones que se corresponden con la técnica de elicitación de brainstorming:

- **Promueve el desarrollo de ideas creativas para obtener soluciones al problema.**
- Involucra activamente a la gerencia en el desarrollo del proyecto. → **Técnica de JRP.**
- **Ayuda a entender el dominio del problema.**
- Requiere extenso entrenamiento de todos los participantes. → **Técnica de JRP.**
- Reduce el tiempo de la etapa de elicitación de requerimientos. → **Técnica de JRP.**

## **AUTOEVALUACIÓN 2 → Especificación de Requerimientos / Historias de Usuario**

1. Seleccione para cada técnica de especificación si es estática o dinámica:

- Expresiones regulares. → **Estática.**
- Abstracciones. → **Estática.**
- Redes de Petri. → **Dinámica.**
- Diagrama de Transición de Estados. → **Dinámica.**
- Relaciones de Recurrencia. → **Estática.**
- Historias de Usuario. → **Dinámica.**

2. Seleccione el tipo de requerimiento no funcional en cada característica del sistema presentado:

- Uso de recursos. → **Rendimiento.**
- Documentación de usuarios. → **Usabilidad.**
- Estética. → **Usabilidad.**
- Facilidad de Recuperación. → **Fiabilidad.**

- Disponibilidad. → **Rendimiento**.

3. Elija para cada definición la característica correcta

- Las historias de usuario cubren requerimientos funcionales, por lo que generalmente son verificables. → **Verificables**.
- Un resultado de la discusión de una historia de usuario es la estimación del tiempo que tomará completarla. → **Estimables**.
- De ser necesario, combinar las historias dependientes o buscar otra forma de dividir las historias de manera que resulten independientes. → **Independientes**.
- La historia en sí misma no es lo suficientemente explícita como para considerarse un contrato, la discusión con los usuarios debe permitir esclarecer su alcance y éste debe dejarse explícito bajo la forma de pruebas de validación. → **Negociables**.

4. Seleccionar cuales de las siguientes afirmaciones representan conceptos de las historias de usuarios (HU):

- Al momento de implementar las historias, los desarrolladores deben programarlas y no discutir las con los clientes.
- Permiten responder abiertamente a los requisitos cambiantes. ✓
- La HU debería poder escribirse sobre una nota adhesiva pequeña. ✓
- Generalmente se espera que la estimación de tiempo de cada historia de usuario se sitúe entre unas 2 horas y una semana.
- Estimaciones mayores a una semana son indicativos de que la historia de usuario es muy compleja y debe dividirse.

5. Seleccione para cada característica si se trata de un beneficio o una desventaja de la HU.

- Fomentan aplazar los detalles no imprescindibles. → **Ventaja**.
- Permite dividir los proyectos en pequeñas entregas. → **Ventaja**.
- Proporcionan la documentación necesaria fomentando a la vez el debate. → **Ventaja**.
- Involucran y captan al cliente para el proceso y para el producto. → **Ventaja**.
- Se requiere contacto permanente con el cliente durante todo el proyecto. → **Desventaja**.
- Se necesitan desarrolladores cualificados que tengan una visión muy clara sobre sus posibilidades y cómo conseguir objetivos a corto plazo. → **Desventaja**.

6. Complete las siguientes oraciones arrastrando la palabra al lugar elegido:

La Ingeniería de requerimientos es el proceso mediante el cual se intercambian diferentes **puntos de vista** para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una **combinación** de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un **documento** de requerimientos denominado **SRS**.

En esta ingeniería se transforman los **requerimientos** declarados por los clientes, ya sean hablados o escritos, a especificaciones precisas, no ambiguas, **consistentes** y completas del **comportamiento** del sistema, incluyendo funciones, **interfaces**, rendimiento y limitaciones.

7. En el SRS hay diferentes usuarios de ese documento, indique a que tipo de usuario se corresponde cada definición:

- Especifican los requerimientos y los leen para ver si cubren sus necesidades. → [Clientes](#).
- Usan requerimientos para hacer pruebas de validación. → [Ingenieros de prueba](#).
- Usan los requerimientos para comprender el sistema y las relaciones de los componentes del sistema. → [Ingenieros de mantenimiento del sistema](#).
- Usan requerimientos para entender el sistema a desarrollar. → [Ingenieros de sistema](#).
- Usan el SRS para planear la cotización del sistema. → [Administradores](#).

8. Seleccionar aquellas afirmaciones que resulten verdaderas para la etapa de validación de requerimientos:

- La validación solo se puede hacer con la activa participación del usuario. ✓
- En la revisión formal de requerimientos los desarrolladores deben tratar los requerimientos con tantos stakeholders como sea posible.
- La validación de requerimientos comprende las verificaciones de realismo (es decir, que los requerimientos se pueden implementar). ✓
- En la etapa de validación se describe el ambiente en el que debe operar el sistema. ✓
- Se puede probar formalmente que un modelo de requerimiento es correcto.

9. Seleccione las afirmaciones que considere se corresponden a un requerimiento funcional.

- Se derivan de las políticas y procedimientos existentes en la organización del cliente.
- Describen lo que el sistema debe hacer o incluso como NO debe comportarse. ✓
- Especifican la usabilidad del producto.
- Se derivan de las políticas y procedimientos existentes en la organización del desarrollador de software (entrega, estándares).
- Se pueden expresar de diferentes formas. ✓
- Describen con detalle la funcionalidad del requerimiento. ✓

10. Seleccione las afirmaciones correctas:

- Un requerimiento permite que los desarrolladores expliquen cómo han entendido lo que el cliente pretende del sistema. ✓
- Un requerimiento conciso es fácil de leer y entender. ✓
- Un requerimiento no indica al equipo de pruebas que demostraciones llevar a cabo.

- Un requerimiento es verificable si su omisión provoca una deficiencia.
- Un requerimiento es consistente si tiene una sola implementación.
- Un requerimiento se considera completo si no es necesario ampliarlo. ✓

### AUTOEVALUACIÓN 3 → Casos de Uso

1. Seleccione la respuesta correcta: los elementos de modelado de un Diagrama de CU son:
  - Casos de Uso – Sistema – Subsistema.
  - Actores – Casos de Uso – Sistema.
  - Actores – Casos de Uso – Relaciones. ✓
  - Actores – Casos de Uso – Subsistemas.
2. Seleccione las respuestas que considere correctas. Los principales propósitos de los CU son:
  - Presentar un orden cronológico de la funcionalidad del sistema.
  - Identificar los requerimientos no funcionales del sistema.
  - Describir los requerimientos funcionales del sistema. ✓
  - Relacionar las funciones del sistema con las clases del sistema.
  - Identificar los roles que interactúan con el sistema. ✓
3. Seleccione los pasos generales que se corresponden con el proceso de modelado de los CU:
  - Construir un Diagrama para el modelo de CU. ✓
  - Crear una épica como un conjunto de CU que se agrupan por algún denominador común.
  - Identificar los CU para los requerimientos. ✓
  - Definir en un Diagrama de contexto el alcance del sistema.
  - Revisar los manuales de usuario de sistemas similares para realizar documentación del sistema a desarrollar.
  - Identificar a los actores. ✓
  - Hacer minutas de las reuniones de definición de CU.
  - Documentar las narraciones de CU para los requerimientos. ✓
4. Seleccione las características que definen a un actor.
  - Modela un tipo de rol que interacciona con el sistema. ✓
  - Puede modelar a una persona y un subsistema, pero no a un dispositivo o al tiempo.
  - Puede modelar a una persona, un subsistema, un dispositivo o al tiempo. ✓
  - Puede modelar al sistema en desarrollo.
  - En interno al sistema.
5. Dado el siguiente problema, seleccione los CU que se corresponden con el actor "usuario".

Sokoban es un juego de varios niveles. Cada nivel está compuesto por un jugador, cajas, repisas y muros. El objetivo del jugador es empujar todas las cajas sobre las repisas. Cuando esto sucede el jugador pasa al siguiente nivel. Para mover una caja, el jugador debe colocarse al lado y empujarla. Si la casilla hacia la que está empujando la caja está libre la caja se moverá. Si el jugador se queda bloqueado, es decir, no puede terminar el nivel, puede reiniciar el nivel perdiendo una vida.

Cuando el jugador pierde todas sus vidas la partida termina.

Se requiere que el sistema debe permitir comenzar una nueva partida y terminarla. El sistema debe permitir mover al jugador y a las cajas y reiniciar el nivel cuando el usuario lo solicite. El sistema deberá almacenar varios niveles y cambiar de nivel cuando el usuario complete el nivel actual.

- Cargar un nivel. → No es un CU porque sólo interviene el sistema. Además, cuando detallamos como cargar un nivel estamos detallando el sistema.
- Terminar la partida. → Tal y como está redactado el enunciado la respuesta es no (no lo pide como requerimiento).
- Iniciar partida. ✓
- Cambiar de nivel. → No es un CU porque solo participa el sistema (no participa ningún actor externo). La única manera que un actor externo tiene de cambiar de nivel es mediante los movimientos que realice.
- Mover jugador. ✓
- Iniciar nivel. ✓

6. Seleccione la palabra que define el concepto presentado:

- Un caso de uso que consiste en los pasos extraídos de otro más complejo para simplificar el caso original y, así, ampliar su funcionalidad. Este tipo de caso de uso hace algo que no hace la funcionalidad del caso de uso original. → Extensión.
- Relación entre un actor y un caso de uso en la que interactúan entre sí. → Asociación.
- Cualquier cosa que necesite interactuar con el sistema para intercambiar determinada información. → Actor.
- Cuando dos o más actores comparten un comportamiento común (en otras palabras, pueden iniciar el mismo caso de uso) lo mejor es extrapolar este comportamiento común y asignarlo a un nuevo actor resumen con objeto de reducir la comunicación redundante con el sistema. → Herencia.
- Un caso de uso que reduce la redundancia entre dos o más casos de uso al combinar los pasos comunes existentes en estos casos. → Uso o Inclusión.

7. Coloque la palabra que define adecuadamente a cada descripción presentada de los componentes de los escenarios de los CU:

- Son los involucrados que se benefician del CU al recibir algo de valor medible u observable → **Actores**.
- Representa la interacción entre el actor y el sistema, sin errores ni condiciones. → **Curso normal**.
- Es una restricción del estado del sistema antes de la ejecución del caso de uso. En general, esto se refiere a otro CU que debe ejecutarse previamente. → **Precondición**.
- Es una restricción del estado del sistema después que el caso de uso ha sido ejecutado con éxito. Esto podría ser datos registrados en una base de datos o un recibo entregado a un cliente. → **Postcondición**.
- Es la descripción del comportamiento si ocurre una excepción o variación del curso típico. → **Curso alternativo**.

8. Seleccione las preguntas que realizaría para poder identificar los "actores" de un sistema:

- ¿Quién o qué recibe salidas del sistema? ✓
- ¿Quién le proporciona entradas al sistema? ✓
- ¿Quién se encargará de las pruebas de aceptación del sistema?
- ¿Necesita el sistema informar al actor de cambios o eventos que hayan ocurrido?
- ¿Existen eventos originados automáticamente en un instante predeterminado? ✓
- ¿Quién realizará el manual de usuario del sistema?
- ¿Se requieren interfaces con otros sistemas? ✓
- ¿Quién determina si el sistema es viable para su desarrollo?

9. Un curso alternativo de eventos en el escenario de CU son las actividades realizadas por el(los) actor(es) y por el sistema con objeto de satisfacer la meta del caso de uso. Se incluyen las interacciones entre el sistema y el actor y las actividades realizadas por el sistema como respuesta a las interacciones.

**FALSO:** Los cursos alternos documentan los comportamientos del CU si ocurre una excepción o una variación del curso típico. Esto puede suceder cuando ocurra un punto de decisión dentro del CU o una excepción que requiera pasos adicionales fuera del alcance del curso típico.

10. Arrastre sobre la imagen los tipos de relaciones que se correspondan al diagrama de casos de uso presentado y que responden al siguiente enunciado.

El paciente puede marcar una consulta, cancelar una consulta y emitir una receta de en la forma de actor pasivo. Además, puede recibir recordatorios como actor pasivo desde el caso de uso enviar recordatorio, el cual podría implementarse como un proceso automatizado del sistema.

El actor "Staff" que generaliza a los actores empleado y médico de la clínica puede cancelar consultas como actor pasivo. Es decir, estos dos actores pueden realizar la operación generalizados en el actor "Staff".

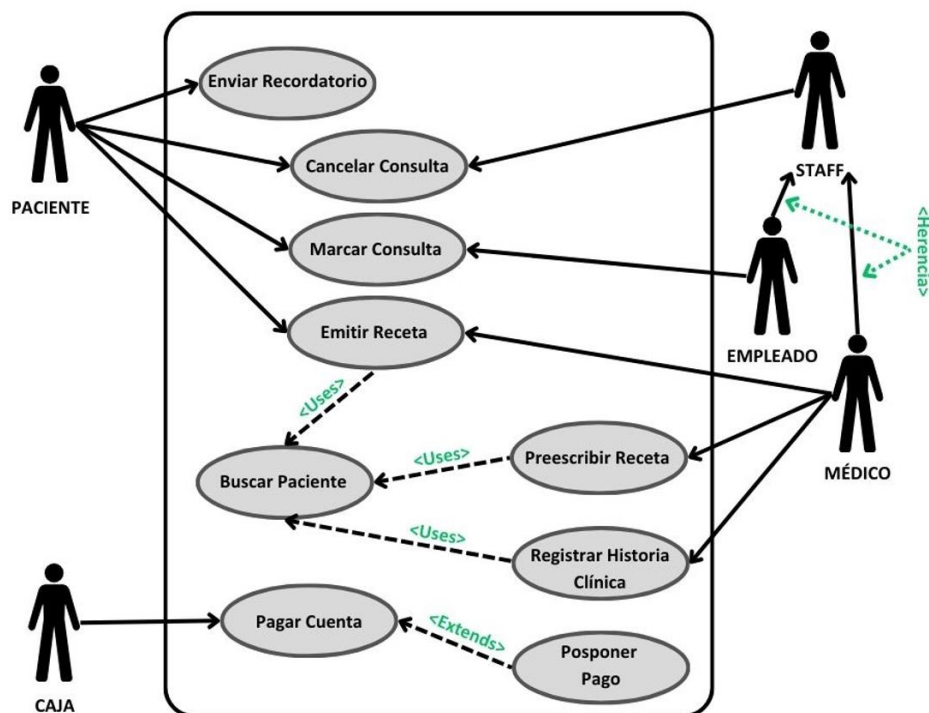
Un empleado sólo puede marcar o cancelar consultas, esta última generalizado en el actor "Staff".

El médico puede emitir receta, prescribir la receta y registrar el historial.

Los casos de uso Emitir receta, Prescribir receta y Registrar historial incluyen el caso de uso "Buscar paciente". Esto podría implementarse como un único proceso que los tres utilizan. De esta forma se sientan las bases para la reutilización de componentes de software en lugar de tener que implementar rutinas de búsqueda de pacientes independientes para los tres.

El actor "Caja" sólo puede pagar una cuenta.

El caso de uso "Pagar cuentas" puede extenderse con "Posponer pago". Observa que a diferencia del caso de uso "Buscar paciente" este no necesita ejecutarse necesariamente, sino solamente en un caso especial o de excepción, en este caso cuando el pago necesita posponerse por alguna razón.



## AUTOEVALUACIÓN 4 → Diagrama de Transición de Estados

### 1. Completar el texto.

Una máquina de estados finitos describe al sistema como un conjunto de **estados** donde el sistema reacciona a ciertos **eventos** posibles **externos** o internos. Una máquina de estado finito



es algo más general que un **diagrama de transición de estados**. Las máquinas de estado finito se modelizan como una **función** aplicada a un estado y un **estímulo** que nos devuelve un nuevo estado de nuestro sistema.

2. Un autómata finito (AF) puede ser descrito como una 5-tupla  $(S, \Sigma, T, s, A)$  Seleccione la definición correcta para cada componente de la 5-tupla:

- $S \rightarrow$  Conjunto de Estados.
- $T \rightarrow$  Función de Transición.
- $\Sigma \rightarrow$  Alfabeto.
- $A \rightarrow$  Conjunto de Estados de Aceptación o Finales.
- $s \rightarrow$  Estado Inicial.

3. Arrastre las palabras para armar una adecuada definición de los Diagramas de Transición y Estados.

Un DTE muestra el **comportamiento** dependiente del **tiempo** de un sistema de información. Representa los **estados** que puede tomar un componente o un sistema y muestra los **eventos** que implican el cambio de un estado al otro.

4. Una transición es una relación entre dos estados que están unidos por una flecha, identificando a una situación particular en un estado primario donde se realizará una acción específica y que pasará a un segundo estado, cuando ocurra un evento y se cumplan unas condiciones específicas. **VERDADERO**.

5. El Diagrama de transición y estados (DTE) es una técnica de especificación estática porque no describe como el sistema responde a eventos.

**FALSO:** El DTE es una técnica de especificación de requerimientos dinámica porque justamente especifica un sistema en función de los cambios que ocurren a lo largo del tiempo.

6. Indique en el siguiente listado cuáles identifica con eventos y cuáles son acciones de un DTE. Teniendo en cuenta que:

- Un evento es una ocurrencia que puede causar la transición de un estado a otro.
- Una acción es una operación atómica, que no se puede interrumpir por un evento y que se ejecuta hasta su finalización.
- Recepción de una llamada para realizar una operación.  $\rightarrow$  Evento.
- Liberar habitación.  $\rightarrow$  Acción.
- Crear un archivo.  $\rightarrow$  Acción.
- Paso de cierto periodo de tiempo, después de entrar al estado actual, o de cierta hora y fecha concretas.  $\rightarrow$  Evento.
- Recepción de una señal.  $\rightarrow$  Evento.

- Modificar la hora. → [Acción](#).
7. Mediante la descripción de las transiciones es posible describir con más detalle las condiciones bajo las cuales se abandona un estado para pasar al siguiente. **VERDADERO.**
8. Ordenar los pasos para crear un DTE:
- Identificar los Estados.
  - Si hay un estado completo se puede explotar.
  - Se identifica el Estado Inicial.
  - Se identifican los cambios de estado con flechas.
  - Se analizan las condiciones y las acciones.
  - Se verifica la consistencia.
9. Según lo visto en teoría identifique para el siguiente problema cuales son estados y cuales transiciones (evento, condición o acción).

La fotocopidora estará apagada durante los fines de semana. Los lunes se enciende y se queda en modo hibernación hasta que llega un usuario a fotocopiar. El usuario debe introducir el dinero y cuando lo considere oportuno pulsa el número de copias que desea hacer, o simplemente pulsa fotocopiar si por defecto quiere hacer una sola copia. Si no hubiera dinero suficiente para hacer las copias requeridas se lo indicará al usuario.

Cuando ha terminado de hacer las copias muestra el saldo disponible. Si el usuario ha terminado de fotocopiar debe poder recuperar el dinero que quede en la fotocopidora.

Los sábados a las 10 de la mañana la fotocopidora se apaga y vuelve a encenderse los lunes a las 7 de la mañana.

- Dinero no suficiente. → [Condición](#).
- Devolver dinero. → [Acción](#).
- Operación completada. → [Evento](#).
- Pidiendo número de copias. → [Estado](#).
- Fotocopidora lista. → [Evento](#).
- Aceptando dinero. → [Estado](#).
- Pulsa copiar. → [Evento](#).
- Encender fotocopidora. → [Acción](#).
- Calentando. → [Estado](#).
- Devolviendo dinero. → [Estado](#).
- Esperando dinero. → [Estado](#).
- Dinero suficiente. → [Condición](#).



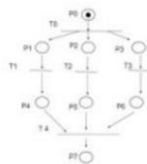
- Es un grafo en el cual desde un nodo puede partir más de una arista o arco de manera que su conjunto de vértices puede particionarse en varios conjuntos independientes donde las aristas tienen un sentido definido.
  - Es un grafo en el cual desde un nodo puede partir hasta tres aristas o arcos de manera que su conjunto de vértices puede particionarse en dos conjuntos independientes donde las aristas tienen un sentido definido.
  - Es un árbol en el cual desde un nodo puede partir varias aristas o arcos de manera que su conjunto de vértices puede particionarse en dos conjuntos independientes donde las aristas tienen un sentido definido.
  - Es un grafo en el cual desde un nodo puede partir más de una arista o arco de manera que su conjunto de vértices puede particionarse en dos conjuntos independientes donde las aristas tienen un sentido definido. ✓
  - Es un grafo en el cual desde un nodo puede partir más de una arista o arco de manera que su conjunto de vértices puede particionarse en dos conjuntos independientes donde las aristas son relaciones simétricas y no apuntan en ningún sentido.
4. En una Red de Petri los estados se representan como transiciones y los eventos como lugares o sitios.

**FALSO:** Es al revés, en una RP, los eventos o acciones se representan como transiciones y los estados o condiciones como lugares o sitios.

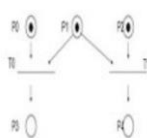
#### 5. Completar las palabras que faltan.

La ocurrencia de un evento está sujeta a que se den ciertas **precondiciones** y al ocurrir el evento causa que se hagan verdaderas las **postcondiciones**. En las Redes de Petri el **orden** en que ocurren los eventos es uno de los permitidos y son **asincrónicas**. La ejecución es **no determinística**. Se acepta que el disparo de una transición es **instantáneo**.

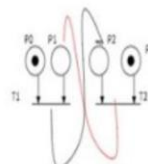
#### 6. Coloque el nombre que corresponde a cada situación modélica de concurrencia representada en Redes de Petri:



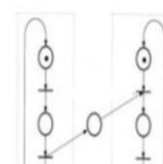
Paralelismo(parallel)



Exclusión mutua .



Deadlock -inanición .



Producir y consumir .

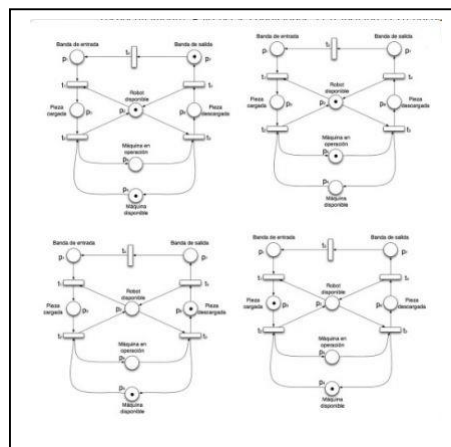
7. Con las Redes de Petri se pueden modelar diferentes situaciones que se producen durante la concurrencia. Indicar para cada definición cual es esa situación que se está modelando.

- Es un bloqueo permanente de un conjunto de procesos o hilos de ejecución en un sistema concurrente que compiten por recursos del sistema o bien se comunican entre ellos. → **DeadLock**.
- Es una forma de computación en la cual varios cálculos/procesos pueden realizarse simultáneamente → **Paralelismo**.
- Estrategia que se utiliza para evitar que entre más de un proceso a la vez en la sección crítica (por ejemplo, un recurso compartido como memoria, o impresora). → **Exclusión mutua**.
- Varios procesos pueden colaborar en la solución de un problema para eso necesitan compartir información y recursos, pero esto debe ser controlado para asegurar la integridad y correcta operación del sistema. → **Sincronización**.

8. Es posible definir la ejecución de una Red de Petri con marcas. Seleccione las reglas del siguiente listado pueden definir la ejecución de la red:

- Una transición está habilitada si cada uno de sus lugares de entrada contiene tantos tokens como arcos haya desde el lugar de entrada a la transición. ✓
- Al disparar una transición no es necesario distribuir todos los token, lo puedo hacer de a uno a la vez en diferentes momentos en diferentes disparos. ✓
- Cada disparo de una transición modifica la distribución de las fichas, y por ello produce un nuevo marcado en la red. ✓
- Una transición está habilitada si tengo al menos un token en uno de los lugares (sitios). ✓
- Una transición habilitada se puede disparar sacando un token de cada lugar de entrada por cada arco que va desde ese lugar de entrada a la transición y colocando un token en cada lugar de salida por cada arco que haya de la transición al lugar de salida. ✓

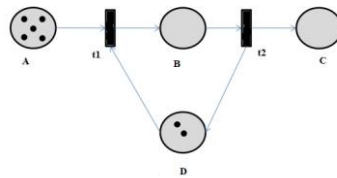
9. Coloque el número de orden de ejecución de esta Red de Petri: **Orden → [4-2-3-1]**



10. Dado el siguiente problema:

Debemos empezar la fabricación de 5 pedidos, sin productos en proceso ni fabricados. Los pedidos en espera deben pasar por el lugar de proceso, y acabar en pedidos completados. No puede haber más de 2 pedidos en proceso al mismo tiempo.

Seleccione la Red de Petri que lo representa:



### AUTOEVALUACIÓN 6 → Tablas de decisión

1. Es una herramienta que permite presentar de forma concisa las reglas lógicas que hay que utilizar para decidir acciones a ejecutar en función de las condiciones y la lógica de decisión de un problema específico. **VERDADERO.**

2. Completar el texto:

Una tabla de decisión describe el sistema como un conjunto de posibles **condiciones** satisfechas por el sistema en un momento dado y **acciones** a ser tomadas como un resultado y de **reglas** para reaccionar ante los estímulos que ocurren cuando se reúnen determinados conjuntos de condiciones.

3. Una especificación contradictoria permite reducir la complejidad de la tabla.

**FALSO:** Las especificaciones redundantes son las que reducen la complejidad de la tabla.

4. Las tablas de decisión se usan para representar la descripción de situaciones, se representan las distintas alternativas, estados de la naturaleza y las consecuencias, proporcionan una descripción completa, correcta, clara y concisa de una situación que se resuelve por una decisión tomada en un momento específico del tiempo. **VERDADERO.**

5. Las tablas de decisión están pensadas y son las más usadas para modelar procesos concurrentes y paralelos.

**FALSO:** Esas son las Redes de Petri.

6. Las condiciones y las acciones en una tabla de decisión pueden ser compuestas donde se utilizan conectivos lógicos.

**FALSO:** Las condiciones o acciones deben ser simples o atómicas.

7. Elegir del siguiente listado las acciones y condiciones que están bien construidas

- Aplicar descuento del 25%. ✓
- Hay stock. ✓
- No es socio.

- Antigüedad mayor a 20 años. ✓
  - Calcular máximo e informar por pantalla el valor.
  - Calcular descuento del 5%. ✓
8. Ordenar los siguientes pasos para construir una Tabla de decisión:
- Identificar condiciones y acciones.
  - Completar la tabla con las condiciones y acciones.
  - Construir las reglas.
  - Revisar completitud de la tabla.
9. Colocar cada definición con el tipo de especificación para una tabla de decisión
- Aquellas que marcan para reglas que determinan las mismas condiciones acciones iguales. → [Redundantes](#).
  - Aquellas que especifican para reglas que determinan las mismas condiciones acciones distintas. → [Contradictorias](#).
  - Aquellas que determinan acción/es para todas las reglas posibles. → [Completas](#).
10. Dado el siguiente problema indique a priori (sin reducir la tabla en ningún sentido) cuantas reglas tiene el siguiente problema:
- Un negocio ofrece los siguientes descuentos: para compras pequeñas (menores a 1000 pesos) no ofrece descuentos, para compras medianas (mayores a 1000 pesos y menores a 10000 pesos) ofrece 5% de descuento y para compras mayores a 10000 pesos ofrece un 10% de descuento. Para los clientes que compran con frecuencia se hace una excepción y se les ofrece un 5% de descuento más. → [16 \(Cantidad de Condiciones \[N = 4\] / Cantidad de reglas \[ \$2^N = 2^4 = 16\$ \]\)](#).

## AUTOEVALUACIÓN 7 → Análisis estructurado moderno y modelos de procesos

1. El análisis estructurado es una actividad de construcción de **modelos**. Mediante una notación creamos modelos que representan los datos y **flujo de información**. Se realizan tres tipos de modelados: de **datos** (por ejemplo, usando DER), de **funciones** del sistema (por ejemplo, usando DFD) y del **comportamiento** del sistema (por ejemplo, usando DTE).
2. Indique del siguiente listado cuáles son los términos equivalentes:
  - Ciclo de vida del Software. ✓
  - Producto de Software.
  - Modelo de proceso. ✓
  - Standard de Software.
  - Proceso.
  - Paradigma de Software. ✓

- Ingeniería de Software.
3. El modelo de desarrollo en fases incremental entrega un sistema completo desde el principio y luego aumenta la funcionalidad de cada subsistema con las nuevas versiones.  
**FALSO:** Ese sería el modelo de desarrollo en fases iterativo. El modelo por fases incremental entrega un sistema con una funcionalidad inicial y luego se van agregando nuevas funcionalidades a medida que se agregan nuevos subsistemas.
  4. El modelo de cascada combina las actividades de desarrollo con la gestión del riesgo.  
**FALSO:** No, el modelo que combina las actividades de desarrollo con la gestión de riesgo es el modelo en espiral.
  5. El modelo en espiral es candidato para aquellos proyectos donde el usuario tiene dificultad al tratar con los modelos gráficos para modelar los requerimientos y el comportamiento.  
**FALSO:** No, de hecho, es un modelo más complejo, los modelos más útiles para explicarle los modelos a los usuarios o clientes son los más sencillos (por ejemplo, el modelo en cascada).
  6. El modelo descriptivo y el modelo prescriptivo de Software deberían ser iguales. **VERDADERO.**
  7. Coloque para cada concepto la palabra que lo define:
    - Herramienta que permite visualizar un sistema como una red de procesos funcionales conectados entre sí por “conductos” y almacenamientos de datos. → [Diagrama de flujo de datos](#).
    - Es una transformación que es aplicada a los datos, o al control y los modifica. → [Proceso](#).
    - Representa la información en reposo utilizada por el sistema independientemente del sistema de gestión de datos. Contiene la información necesaria para la ejecución del proceso. → [Almacén](#).
    - Representa uno o más elementos de datos (objetos de dato). → [Flecha](#)
    - Es una definición sin ambigüedad de los datos y elementos del sistema. → [Diccionario de datos](#).
    - Es un elemento del sistema (por ejemplo, un elemento hardware, una persona, otro programa) u otro sistema que produce información para ser transformada por el software, o recibe información producida por el software. → [Entidad externa](#).
  8. ¿Cuál de los siguientes modelos de desarrollo de Software enfatiza la identificación y evaluación de riesgos en cada iteración? → [Modelo en espiral](#).
  9. ¿Cuál es el propósito principal del modelo de prototipo desechable?
    - Probar y validar requisitos antes del desarrollo completo. ✓
    - Integrar funcionalidades incrementales.
    - Reemplazar la fase de diseño final.



- Facilitar el mantenimiento del software.

**10.** En el modelo en V, ¿En qué fase se lleva a cabo la verificación de los requisitos?

- Fase de validación.
- Fase de desarrollo. ✓
- Fase de análisis.
- Fase de diseño.

## **AUTOEVALUACIÓN 8 → Metodologías Ágiles**

**1.** Indique del siguiente listado de características de metodologías de desarrollo cuáles son las que se corresponden con metodologías ágiles y cuales con metodologías no ágiles.

- Énfasis en los aspectos humanos: el individuo y el trabajo en equipo. → Ágil.
- No existe un contrato tradicional, debe ser bastante flexible. → Ágil.
- Se esperan cambios durante el proyecto. → Ágil.
- La arquitectura se va definiendo y mejorando a lo largo del proyecto. → Ágil.
- El cliente es externo al equipo e interactúa con el equipo de desarrollo mediante reuniones. → No ágil.
- Más Roles, más específicos. → No ágil.
- Metodologías aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos. → No ágil.
- Siguen un enfoque lineal o del tipo de lo más general a lo más particular. → No ágil.
- Pocos Artefactos. El modelado es prescindible, modelos desechables. → Ágil.

**2.** Indique cuales de los siguientes enunciados son valores establecidos por las metodologías ágiles:

- Procesos y herramientas más que individuos e interacciones.
- Software operante más que documentaciones completas. ✓
- Negociaciones contractuales más que colaboración con el cliente.
- Respuesta al cambio más que apegarse a una rigurosa planificación. ✓

**3.** ¿Cuál de los siguientes describe una ventaja del enfoque Lean en Kanban?

- Establece roles rígidos y secuenciales en el proyecto.
- Limita la flexibilidad del equipo.
- Aumenta la cantidad de documentación del proyecto.
- Fomenta la mejora continua y la reducción de desperdicios. ✓

**4.** Indique cuales de los siguientes son metodologías ágiles:

- XP (eXtreme Programming). ✓
- Modelo en V.
- Modelo es espiral de Bohem.
- Prototipado desechable.
- Desarrollo de Software adaptativo (ASD). ✓
- SCRUM. ✓
- Desarrollo por fases incremental.

5. Indique cuales son los valores [**principios básicos**] del XP:

- Testing.
- Complejidad.
- Retroalimentación. ✓
- Refactoring.
- Coraje. ✓
- Propiedad colectiva del código.
- Comunicación. ✓
- Respeto. ✓
- Simplicidad. ✓

6. Coloque para cada función el rol que se corresponde en XP:

- El **programador** es el responsable de construir el sistema.
- El **jeje de proyecto (manager)** es quien organiza y guía las reuniones.
- El **rastreador (tracker)** conserva los datos históricos.
- El **entrenador (coach)** es el responsable del proceso.
- El **cliente (customer)** establece las pruebas funcionales.

7. Indique de las siguientes afirmaciones cuáles son principios de las metodologías ágiles (MA):

- Se sugiere que no trabajen juntos a lo largo del proyecto la gente del negocio y los desarrolladores dado que tienen intereses diferentes.
- Los procesos ágiles promueven un desarrollo sostenible. ✓
- Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva. ✓
- La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor. ✓
- Construir proyectos que no tengan en cuenta las motivaciones individuales de los participantes, lo importante es la gestión.
- El software que funciona no es la medida clave de progreso.

- El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo. ✓
  - Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas. ✓
8. Indique de la siguiente lista los artefactos de la metodología ágil SCRUM.
- Product backlog. ✓
  - Exploración.
  - Product owner.
  - Sprint backlog. ✓
  - Burndown chart. ✓
  - Planificación.
9. Indicar de la siguiente lista los principios de la metodología ágil SCRUM.
- Crear conocimiento. En la práctica no se puede tener el conocimiento antes de empezar el desarrollo. ✓
  - Eliminar el desperdicio: no generar artefactos, ni perder el tiempo haciendo cosas que no le suman valor al cliente. ✓
  - Entregar rápido. ✓
  - Tomar las decisiones en el momento adecuado, esperar hasta ese momento. Si se puede esperar, mejor. ✓
  - Optimizar la etapa de generación de product backlog (lista maestra), el resto no es necesario optimizar.
  - Construir la calidad separada del producto, la calidad es inyecta en una etapa final.
  - Documentar. Generar la mayor cantidad posible de documentos que acompañen a todo el proceso, aunque este se atrase. Generar manuales de usuario para cada función del sistema.
  - Entregar en el momento adecuado luego de realizar al menos tres sprint.
10. ¿Cuál es la práctica de KANBAN?
- Visualizar el flujo de trabajo. ✓
  - Limitar el trabajo en curso. ✓
  - Evitar los cambios en los requisitos.
  - Prescribir roles específicos.

## AUTOEVALUACIÓN 9 → MDD y Calidad

1. Para cada definición indicar qué tipo de modelo se corresponde:

- Un **PIM** es un modelo de un sistema que no contiene información acerca de la plataforma o la tecnología que es usada para implementarlo.
  - **Transformación de modelos** especifica el proceso de conversión de un modelo en otro modelo del mismo sistema
  - **PSM** es un modelo de un sistema que incluye información acerca de la tecnología específica que se usará para su implementación sobre una plataforma específica.
2. Indique según lo visto en teoría cuales son los beneficios del MDD (Desarrollo de software dirigido por modelos):
- Adaptación a los cambios tecnológicos. ✓
  - Solapamiento de fases en entregas iterativas con testeo de clientes.
  - Los clientes forman parte del equipo de desarrollo.
  - Desarrollo centrado en los clientes.
  - Re-uso (de modelos y transformaciones). ✓
  - Poca automatización y mayor intervención de programación.
  - Automatización. ✓
  - Adaptación a los cambios de requisitos. ✓
  - Tomar las decisiones tecnológicas al principio del desarrollo.
  - Posibilidad de demorar decisiones tecnológicas. ✓
3. Indique para cada norma o modelo qué calidad evalúa:
- CMMI → Calidad del proceso de desarrollo.
  - ISO/IEC 12207 → Calidad del proceso de desarrollo.
  - COMPETISOFT → Calidad del proceso de desarrollo.
  - ISO/IEC 20000 → Calidad de servicios.
  - ISO/IEC 25000 → Calidad de producto de Software.
  - ISO 9001 → Calidad de procesos/servicios y productos en general.
4. Según la "Visión holística de la calidad" de Stylianou y Kumar seleccione el tipo de calidad que se corresponde con las diferentes tareas de un sistema de información:
- Presupuesto y planificación. → Calidad de la Gestión.
  - Datos de calidad. → Calidad de Información.
  - Datos que se ingresan al sistema de información. → Calidad de Datos.
  - Redes y sistemas de software. → Calidad de Infraestructura.
  - Atención al cliente. → Calidad de Servicio.
  - Aplicaciones de software. → Calidad de Software.

5. Seleccione del siguiente listado el o los criterios que considere que definen adecuadamente el concepto de calidad en general:

- La calidad es intangible y por lo tanto no mensurable.
- Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor.  
"Este tapizado del auto es de buena calidad. ✓
- La calidad se origina en el departamento de calidad.
- Su significado sigue siendo ambiguo y muchas veces su uso depende de lo que cada uno entiende por calidad, por lo cual es importante comenzar a unificar su definición. ✓
- Los problemas de mala calidad son originados por los trabajadores de producción.
- Un producto de calidad es un producto de lujo.

6. Seleccione cuáles de las siguientes características del producto de software son las que indica el modelo 25010 que deben ser tenidos en cuenta, según lo visto en teoría:

- Seguridad. ✓
- Experiencia de usuario.
- Portabilidad. ✓
- Eficacia.
- Correctitud.
- Ayuda en línea.
- Facilidad de uso. ✓

7. El modelo de calidad del CMMI escalonado centra su foco en la madurez de la organización y se compone de 6 (seis) niveles.

**FALSO:** Son 5 niveles (Inicial, gestionado, definido, gestionado cuantitativamente, optimizado).

8. El modelo de calidad de la 25010 define medidas de calidad, define cómo mido las características del software que indica la 25020.

**FALSO:** Es al revés, la 25010 define las características y la 25020 define cómo medirlas

9. La estandarización del proceso de software define las propiedades que debe satisfacer el producto software resultante.

**FALSO:** No, lo que define es como llevar a cabo las actividades de desarrollo de Software, pero no define las propiedades del producto de Software resultante.

10. La sigla SWEBOK corresponde a una norma de calidad.

**FALSO:** No es una norma de calidad, sino un conjunto de buenas prácticas.