

Actividad 1 – Modelado de escenarios de casos de uso

Lenguaje de modelado unificado

Ingeniería en Desarrollo de Software

Tutor:

Eduardo Israel Castillo García

Alumno:

Alejandro Abarca Gerónimo

Fecha:

18 de abril de 2023

Indice

Introducción.	3
Descripción.	4
Justificación.	5
Desarrollo.	6
Definición de requisitos funcionales y no funcionales.	6
Casos de uso.	7
Ficha de especificaciones.	8
Conclusión.	9
Referencias.	9

Introducción.

El lenguaje de modelado unificado está muy relacionado con los diagramas de casos de uso ya que por medio de estos se definen los procesos que un determinado sistema va ejecutar. El lenguaje está compuesto por una notación muy específica y por las reglas semánticas relacionadas para la construcción de sistemas de software.

El modelo de casos de uso describe la funcionalidad propuesta de los nuevos sistemas y un Caso de Uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema. Un Caso de Uso es una unidad de trabajo significativo; por ejemplo crear una solicitud y modificar una solicitud son todos Casos de Uso.

Cada Caso de Uso tiene una descripción que especifica la funcionalidad que se incorporará al sistema propuesto. Un Caso de Uso puede 'incluir' la funcionalidad de otro Caso de Uso o puede 'extender' otro Caso de Uso con su propio comportamiento.

Los casos de uso se relacionan con 'actores'. Un actor es un humano o una máquina que interactúa con el sistema para realizar un trabajo significativo.

La especificación formal de un Caso de Uso incluye 3 elementos básicos:

1. **Requisitos:** son los requisitos funcionales formales que el Caso de Uso debe proveer al usuario final. Estos corresponden a las especificaciones funcionales de las metodologías estructuradas. Un requisito es un requerimiento, algo necesario para que el Caso de Uso pueda dar como resultado alguna acción o proveer algún valor al sistema.

2. **Restricciones:** estas son las reglas formales y las limitaciones bajo las que opera un Caso de Uso e incluyen las pre-condiciones, las post-condiciones y las invariantes.
 - **Precondición** especifica qué debe haber ocurrido o estar cumplido antes de que el Caso de Uso pueda iniciarse.
 - **Post-condición** documenta qué será verdadero una vez que el Caso de Uso se complete.
 - **Invariante** especifica qué será verdadero durante el tiempo en que opere el Caso de Uso.
3. **Escenarios:** son descripciones formales del flujo de eventos que ocurre durante una instancia de un Caso de Uso. Se describen con texto y corresponden a una representación textual del diagrama de secuencia.

Descripción.

Para el desarrollo de esta actividad se nos pide analizar el contexto proporcionado en el PDF de la actividad el cual trata de un sorteo llamado “Vacaciones en familia”. Para ello se nos pide analizar detenidamente ya que se nos presentan pre-condiciones que se deben cumplir para llegar al Caso de uso. También se nos presentan post-condiciones, estos segundos especifican si un terminado proceso se cumple o no dentro de los mismos Casos de uso.

Los puntos a considerar son los siguientes y en base a ello desarrollaremos dicho diagrama de casos de uso:

- El cliente será acreedor de un boleto participante, por cada \$500 pesos en transacciones participantes.
- Al cliente se le entregará el ticket de transacción, si la transacción cumple con el primer requisito (por cada \$500) se le otorgará un ticket de boleto

participante y ticket ganador o perdedor según el caso, en caso de ser una transacción que no participa al cliente se le entregará el ticket de transacción.

- No participarán empleados.
- Cada 2000 boletos habrá un boleto ganador.
- Habrá 1000 boletos ganadores.
- Si el boleto resulta ganador deberá responder una pregunta de opción múltiple con 3 posibles respuestas.
- Cada respuesta errónea se habilitará hasta que se seleccione la respuesta correcta.
- Se validará que el sorteo esté activo o dentro del rango de fecha.

Justificación.

Considero que el diagrama realizado cumple y ejecuta las necesidades del sistema y usuarios ya que se han considerado todas las especificaciones que se nos han proporcionado. Es un diagrama robusto, claro y basto que resuelve lo planteado en el contexto y que al analizarlo de manera gráfica logra ser claro y entendible.

Las principales ventajas de utilizar este diagrama son:

- La captura de los requisitos funcionales desde el punto de vista del usuario.
- La utilización de los casos de uso para educir y documentar los requisitos funcionales.
- El manejo de la complejidad en sistemas robustos, descomponiendo el problema en funciones más simples.

La obtención del diagrama de casos de uso es importante para determinar los requisitos necesarios para que funcione el sistema puesto que, si se puede acortar el tiempo en la elaboración de estos diagramas, una aplicación de software se podría

conceptualizar en un tiempo menor. El principal interés en estos trabajos, radica en la obtención de los casos de uso, los actores y las relaciones de asociación.

Desarrollo.

Definición de requisitos funcionales y no funcionales.

1. Requisitos funcionales:

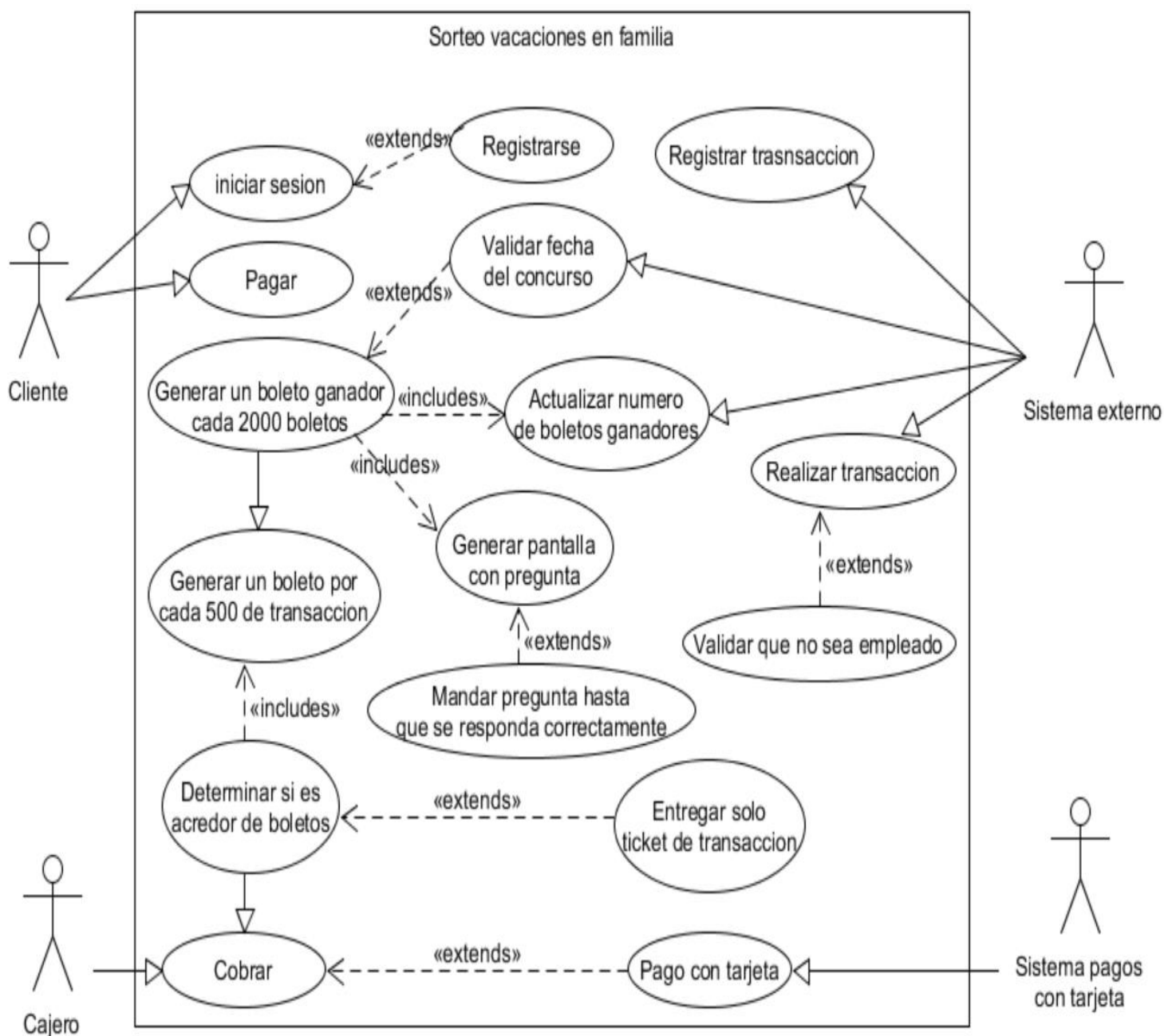
- Descripción de la información que se va ingresar en el sistema.
- Descripción de las operaciones que realizará cada pantalla.
- Descripción de los reportes del sistema.
- Definición de quien puede ingresar datos en el sistema.
- Como el sistema cumplirá los reglamentos y regulaciones generales que le sean aplicables.
- Capturar los datos de la transacción a realizar.
- Interfaz del sistema (botones, colores, formas y acciones).
- El sistema mostrará la pantalla con la pregunta con respuestas de opción múltiple una vez que la transacción del cliente sea aprobada. Para ello el cliente tendrá la opción de responder las 3 veces que se le dan al colocar las 3 posibles respuestas.

2. Requisitos no funcionales:

- El sistema será capaz de identificar cuando un empleado realice una transacción ya que estos nos participan en el sorteo.
- El sistema será capaz de identificar si la transacción del cliente participa en el sorteo o no.
- El sistema emitirá un ticket (boleto) si la transacción cumple con la cantidad establecida en el sorteo.

- El sistema emitirá un boleto ganador cada 2000 transacciones realizadas teniendo un total de 1000 boletos ganadores.
- El sistema hará una verificación sobre el sorteo verificando que se encuentre activo y dentro de la fecha establecida (vigencia del sorteo).

Casos de uso.



Ficha de especificaciones.

- Nombre: Sorteo vacaciones en familia.
- Autor: Alejandro Abarca Gerónimo.
- Fecha: 18 de abril de 2023.
- Descripción: Se nos pide realizar un diagrama de C.U para hacer funcional el sistema del sorteo y obtener a los clientes ganadores. Para ello se han establecido diferentes actores que serán participes dentro de la creación del sistema.
- Actores:
 - ✓ Cliente.
 - ✓ Cajero.
 - ✓ Sistema externo.
 - ✓ Sistema de pagos con tarjeta.
- Precondiciones:
 - ✓ El cliente será acreedor de un boleto participante, por cada \$500 pesos en transacciones participantes.
 - ✓ Si la transacción participa entregar cantidad de boletos dependiendo de la cantidad de la transacción sino entregar solo el ticket de transacción.
 - ✓ Los empleados no participan.
 - ✓ Entregar un boleto ganador cada 2000 boletos emitidos.
 - ✓ Mandar las respuestas a la pregunta realizada hasta que se conteste correctamente.
- Flujo normal: el cliente llega al área de caja e inicia la transacción en conjunto con el cajero, se procesa la información, se procede a realizar el pago y se selecciona el método de pago. El sistema procede a validar si el cliente es acreedor de boletos en base a la transacción realizada, se emiten los boletos en caso de aplicar sino solo el ticket de transacción. Termina.
- Flujo alternativo: el cliente cancela la transacción (solo aplican compras realizadas) y se cancelan los boletos generados y emitidos de forma automática, se restablece el la cantidad de boletos ganadores disponibles y se actualiza el número de boletos emitidos (total de boletos emitidos por transacción realizadas). Termina.
- Postcondiciones:
 - ✓ Emitir tickets de transacciones y boletos.
 - ✓ Cuando termina la ejecución del caso de uso el sistema de vuelve la pantalla inicial para atender al siguiente cliente.

Conclusión.

La importancia de la implementación de los diagramas de casos de uso es muy eficaz al momento de realizar el análisis de los problemas planteados ya que por medio de estos nos damos una idea de la representación gráfica que se tendrá dentro del sistema así como también las funcionalidades del sistema y quien realizara cada una de las tareas representadas en los casos de uso.

Es de vital importancia la implementación y ejecución de los mismos para detectar cualquier posible error dentro del sistema. Con el desarrollo de esta primera actividad aprendimos el funcionamiento de estos diagramas implementando el desarrollo y creación de uno bajo un contexto diferente a las actividades realizadas en materias anteriores (otros periodos). Aprendimos las funciones de la conexión *includes* y *extends* y en qué casos se pueden usar. Utilizamos la herramienta UMLet para la creación del diagrama así como también se realizó una investigación para crear el contenido de este documento.

Referencias.

1. Herramientas UML para el modelado y desarrollo de software - Herramienta de modelado UML de Enterprise Architect. (s. f.). <http://www.sparxsystems.com.ar/>
2. Zapata, C. (s. f.). GENERACIÓN DEL DIAGRAMA DE CASOS DE USO A PARTIR DEL LENGUAJE NATURAL O CONTROLADO: UNA REVISIÓN CRÍTICA. http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0012-73532009000300020#:~:text=RESUMEN%3A%20El%20diagrama%20de%20casos,la%20complejidad%20de%20sistemas%20robustos
3. pmoinformatica.com. (s. f.). Requerimientos funcionales: Ejemplos. <http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>
4. Definición de términos: Caso de uso. (s. f.). https://calidad-software.agesic.gub.uy/MCS-core/guidances/termdefinitions/caso_de_uso_906987BC.html#:~:text=Precondiciones%3A%20Describe%20condiciones%20que%20debe,todo%20el%20caso%20de%20uso