

Actividad | 1 | Modelado de escenarios Casos de Uso

Nombre del curso

Ingeniería en Desarrollo de Software



TUTOR: Miguel Ángel Rodríguez Vega.

ALUMNO: Uziel de Jesús López Ornelas.

FECHA: 01 de Enero del 2025.

Índice

| | |
|---|---|
| Introducción | 1 |
| Descripción..... | 1 |
| Justificación..... | 2 |
| Desarrollo | 3 |
| Definición de requisitos funcionales y no funcionales | 3 |
| Diagrama de Caso de uso | 4 |
| Ficha de especificaciones | 4 |
| Conclusión..... | 5 |
| Referencias..... | 7 |

Introducción

La materia de Lenguaje de Modelado Unificado está a punto de comenzar y con ella aprenderemos aún más sobre las reglas que está presenta, antes que nada, es importante mencionar que un conocimiento base es fundamental para la correcta comprensión de la materia, aunque, en este material tendremos la cortesía de mostrar lo más importante para comprender mejor lo que se verá a continuación. Para entender mejor este concepto daremos algo de contexto, principalmente en lo siguiente: “el lenguaje de Modelado Unificado” es un lenguaje cuyo vocabulario y reglas se centran en la representación conceptual y física de un sistema. Existe una combinación especial que “Unifica” la relación, “Lenguaje = Notación + Reglas (sintácticas, semánticas). En este sentido, UML por sus siglas ofrece un vocabulario y reglas para crear y leer modelos bien formados que constituyen los planos de un sistema de Software. Se le recomienda al lector que mantenga mucha atención a lo que se presentara a continuación.

Descripción

El meta modelo de UML, incluye tres tipos de elementos principales:

- **Bloques de construcción.**
- **Reglas.**
- **Mecanismos.**

Los bloques de construcción tienen diferentes componentes, como lo son:

- **Elementos.**
- **Relaciones.**
- **Diagramas.**

Los **elementos** son los bloques básicos de construcción de un sistema orientado a objetos. Se trata de abstracciones que constituyen los ciudadanos de primera clase de un modelo. Se utiliza para construir modelos bien formados.

Las **relaciones** son las conexiones entre elementos estructurales. Existen cuatro tipos de relaciones y dos tipos especiales de asociación.

Los **diagramas** sirven para visualizar un sistema desde diferentes perspectivas. Se trata de una vista resumida de los elementos que constituye el sistema. Un diagrama puede contener cualquier combinación de elementos y relaciones. Surgen así, los tipos de diagramas de UML 2.

Las **reglas** se encargan de dictar como pueden combinarse los bloques.

Los mecanismos son muy comunes y se aplican a lo largo del lenguaje. Se dividen en:

- **Especificaciones.**
- **Adornos.**
- **Divisiones comunes.**
- **Mecanismos de extensibilidad.**

Los **mecanismos** dan cohesión y simplifican el modelo. Se aplican a todos los bloques de construcción del modelo.

Justificación

A continuación, mencionaremos varias ventajas que tiene UML en este campo:

- **Es estándar.**
- **Facilita la comunicación.**
- **Está basado en un meta modelo con una semántica bien definida.**
- **Se basa en una notación gráfica concisa y fácil de aprender y utilizar.**
- **Es fácilmente extensible.**
- **Se puede utilizar para modelar sistemas de software en diversos dominios.**

Al igual que todo también existen inconvenientes o desventajas, por ejemplo:

- **No es una metodología. Además de UML, hace falta una metodología OO.**
- **No cubre todas las necesidades de especificación de un proyecto de software.**
- **No define los documentos textuales o el diseño de interfaces del usuario.**
- **Faltan elaborados en la documentación.**
- **Puede resultar complejo alcanzar un conocimiento completo del lenguaje.**

Es importante mencionar que un diagrama de clases de UML es una vista del modelo estático del sistema en cuestión. Esto quiere decir que una misma clase puede aparecer en varios diagramas y que podemos crear más de un diagrama.

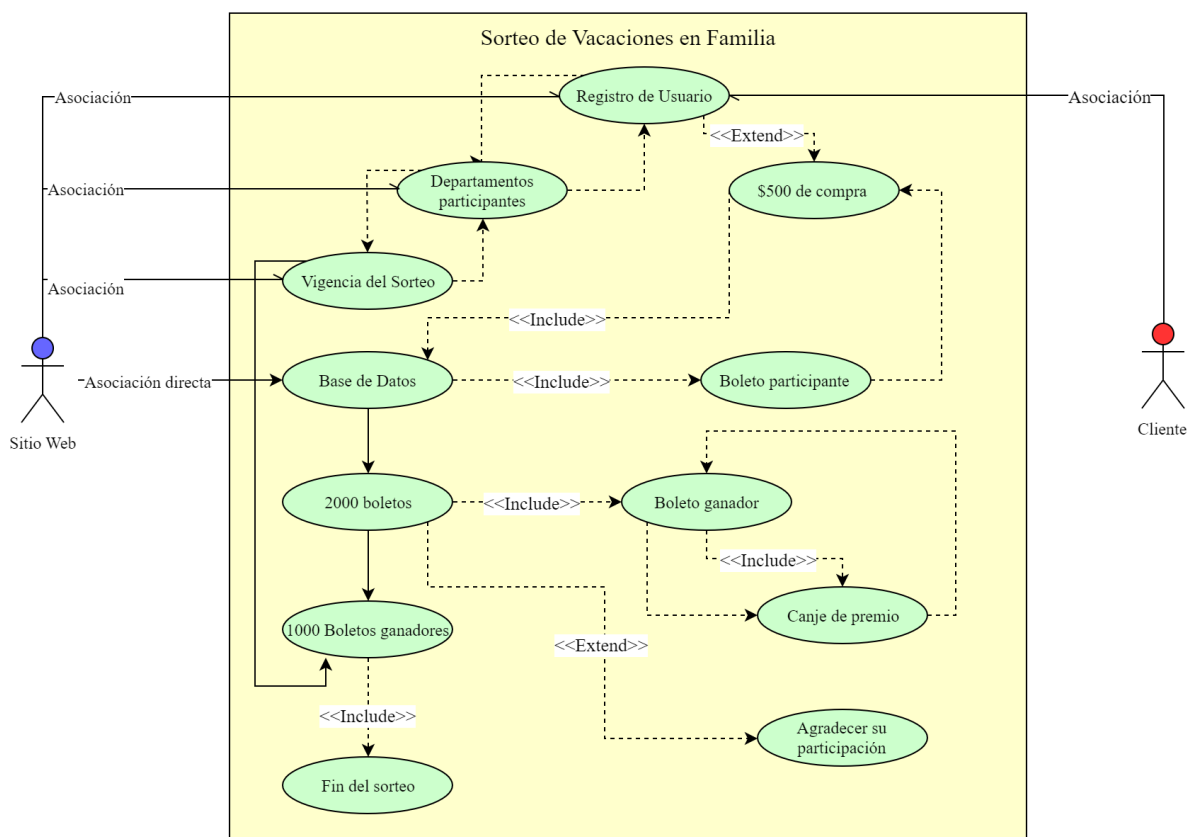
Desarrollo

Definición de requisitos funcionales y no funcionales:

De acuerdo a la contextualización tenemos los siguientes elementos que nos pueden ayudar a la hora de crear nuestro Diagrama de Casos de Uso, entre ellos tenemos los siguientes:

- **Los clientes para poder participar deberán estar registrados en el sitio (esto quiere decir que tendremos dos actores, el cliente y el sitio web).**
- **La vigencia del sorteo y los departamentos participantes son casos en los que se tendrán que tomar en cuenta para la relación del Diagrama.**
- **Por cada “\$500” de alguna compra en los departamentos correspondientes el cliente será candidato para obtener un boleto participante (el monto y el número de boletos que obtiene el cliente también se denota como un caso).**
- **Cada 2000 boletos participantes habrá un ganador (el número de boletos participantes y el ganador en este rubro son otros casos).**
- **Cuando un boleto sea ganador se le indica al cliente en el momento de la compra mostrándole el proceso de canje, si es caso contrario solo se agradece su participación (boleto ganador es mostrado en la compra, caso contrario solo se agradece su participación, son casos de uso)**
- **Solo hay 1,000 boletos ganadores.**
- **Al concluir la vigencia del sorteo y/o la cantidad de boletos ganadores, el sorteo se dará por concluido (se da por finalizado cuando se acabe la vigencia o se acaben la cantidad de boletos ganadores).**

Diagrama de Caso de uso:



Ficha de especificaciones:

La ficha de especificaciones está representada de la siguiente manera:

- Nombre: Sorteo de Vacaciones en Familia.
- Actores: Cliente y Sitio Web.
- Precondiciones del sitio Web: Tiene una asociación con el registro del usuario, departamentos participantes y la vigencia del sorteo con una asociación directa hacia la base de datos en donde se coteja la información.
- Precondiciones del cliente: Tiene una asociación con el registro del sistema en donde se relacionan o unen los dos actores.

- Flujo Normal: en \$500 de compra la acción es la base de datos cotejando la información para brindar el boleto participante, al llegar a 2000 boletos se selecciona a un ganador que al momento se le indicara como canjear su premio, solo hay 1000 boletos ganadores de los cuales si se agotan o la vigencia del torneo termina se da por concluida.
- Flujo alternativo: Al registrarse en la página el cliente tiene la opción de comprar algo de \$500, mas no es obligatorio, al llegar al boleto 2000 hay un ganador de lo contrario se le agradece al cliente su participación.
- Postcondiciones: al ejecutar las precondiciones anteriores se imprimen los resultados que manejan los flujos normales y alternativos presentados anteriormente, ejemplo, en \$500 de compra la acción es la base de datos cotejando la información para brindar el boleto participante, al llegar a 2000 boletos se selecciona a un ganador que al momento se le indicara como canjear su premio, solo hay 1000 boletos ganadores de los cuales si se agotan o la vigencia del torneo termina se da por concluida y de manera secundaria, Al registrarse en la página el cliente tiene la opción de comprar algo de \$500, mas no es obligatorio, al llegar al boleto 2000 hay un ganador de lo contrario se le agradece al cliente su participación.

Conclusión

En esta actividad nos adentramos en la ejecución del diagrama de casos de uso, que es bastante recomendable para lograr un resultado analizando las diferentes acciones que tienen diferentes actores en los que interactúan para relacionarse con flujos directos como secundarios, mostrando de manera gráfica símbolos que representan distintos parámetros establecidos de manera predeterminada. En el documento se realizó el ejemplo de los que es un sorteo y se mostraban aquellos elementos que interactuaban entre sí para lograr una ejecución de cómo se supone que funcionaria dicho proyecto o actividad, que para ser sinceros esto no solo sirve para una actividad o ejercicio, sino para muchas cosas, inclusive puedes crear una rutina diaria con un diagrama de casos de uso en donde observas los diferentes escenarios con los que

puedes interactuar así como las decisiones que elegirían para tener un mejor desempeño y así mejorar el estilo de vida actual, vamos que esto es algo muy práctico.

Link para GitHub:

<https://github.com/Levzu-Ing/Lenguaje-Unificado-de-Modelado.git>

Referencias

Qué es el lenguaje unificado de modelado (UML). (s. f.). Lucidchart.

<https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>