

Actividad | 3 | Modelado UML

Lenguaje Unificado de Modelado

Ingeniería en Desarrollo de
Software



TUTOR: Eduardo Israel Castillo García.

ALUMNO: Sarahi Jaqueline Gómez Juárez.

FECHA: lunes, 16 de septiembre de 2024.

Índice

Índice.....	2
Introducción:	3
Descripción:	9
Justificación:.....	16
Desarrollo.....	19
Contextualización:	19
<i>Diagrama de Componentes UML: “SORTEO EN VACACIONES EN FAMILIA”:</i>	<i>19</i>
<i>Diagrama de Actividades 1: Establecimiento Condiciones y Términos:</i>	<i>22</i>
<i>Diagrama de Actividades 2 con carriles UML: Verificación de Inicio de Sesión</i>	<i>29</i>
<i>Diagrama de Actividades 3 con Carriles UML: VERIFICACION DE TRANSACCIONES Y GESTION DE BOLETOS.....</i>	<i>34</i>
Conclusión:	38
Referencias:	42

Introducción:

En el presente proyecto podrás ver la tercera etapa del Modelado UML (Unified Modeling Language) aplicado en el desarrollo del sistema de gestión del “**Sorteo Vacaciones en Familia**” en una tienda departamental, existen diversos tipos de Diagramas UML, en este documento nos centraremos en dos de ellos: **El Diagrama e Actividades y El Diagrama de Componentes**, seguirá las siguientes directrices descritas dentro de este enlace en su apartado de “**Contextualización**”: https://agcollege.edu.mx/literaturas/59/19/Actividad_1_Lenguaje_Unificado_de_Modelado_V5.docx.pdf

El **Modelado UML (Unified Modeling Language)** es una metodología estandarizada para visualizar, especificar, construir y documentar los componentes de un sistema de software, UML proporciona un conjunto de diagramas que permiten representar tanto la estructura como el comportamiento de un sistema en diversas fases de su desarrollo, es ampliamente utilizado en Ingeniería de Software para diseñar y comunicar la arquitectura de un sistema de manera clara y precisa, existen diferentes **Tipos de Diagramas UML**:

Diagramas estructurales: Describen la organización y estructura del sistema, incluyendo relaciones entre objetos y componentes.

Diagrama de Clases: Muestra la estructura de un sistema a través de sus clases, atributos, métodos y relaciones, descrito a detalle dentro de la segunda etapa del sistema de gestión del “**Sorteo Vacaciones en Familia**” en una tienda departamental. Etapa 2:

<https://drive.google.com/file/d/1bGShAA-tVIFWOW2XcefIBnCGzUB8mEq/view?usp=sharing>

Diagrama de Componentes: Representa los componentes de software y sus

interacciones.

Elementos clave de un diagrama de componentes:

Componentes: Los componentes son bloques modulares que representan partes del sistema, como archivos, bibliotecas, módulos o servicios, se dibujan como rectángulos con dos pequeñas pestañas en el lado izquierdo (una especie de símbolo de “cajón de archivo”).

Pueden representar:

Clases o módulos: Componentes lógicos de software.

Bibliotecas: Un conjunto de clases o servicios empaquetados.

Subsistemas: Un conjunto de módulos o componentes que trabajan juntos.

Interfaces:

Son puntos de interacción entre componentes, donde estos intercambian información o servicios.

Se dibujan como pequeños círculos conectados a los componentes, que pueden representar interfaces proporcionadas (lo que el componente ofrece) o interfaces requeridas (lo que el componente necesita de otros).

Ejemplo: Un componente de “**Validación de Transacciones**” puede tener una interfaz que proporciona los servicios de validación, mientras que un componente de “**Base de Datos**” puede tener una interfaz requerida que necesita acceso a la información de transacciones.

Relaciones de dependencia: Se representan mediante líneas punteadas con una flecha y muestran que un componente depende de otro para funcionar.

Ejemplo: Un módulo “**API de pagos**” podría depender del “**Sistema de autenticación**” para verificar usuarios antes de procesar los pagos.

Relaciones de ensamblaje (Assembly) y delegación:

Relaciones de ensamblaje: Indican que un componente utiliza una interfaz proporcionada por otro componente, se muestran como una línea sólida que conecta las interfaces proporcionadas y requeridas.

Relaciones de delegación: Indican que un componente delega parte de su trabajo a otro componente, a menudo dentro del mismo sistema.

Artefactos: Son objetos físicos como archivos ejecutables, bibliotecas o bases de datos que se relacionan con componentes, se dibujan como rectángulos con una etiqueta en la parte superior izquierda indicando que es un artefacto, los artefactos pueden estar asociados con componentes para mostrar que un componente lógico está implementado en un archivo o recurso físico.

Usos y Beneficios de los Diagramas de Componentes:

Arquitectura modular: Permiten visualizar la estructura física del sistema, destacando los módulos o componentes principales y sus dependencias.

Interfaces claras: Muestran cómo los distintos módulos se comunican a través de interfaces definidas, ayudando a mantener una arquitectura flexible y mantenible.

Gestión de la complejidad: Son ideales para sistemas grandes y complejos, ya que permiten dividir un sistema en componentes más pequeños, lo que facilita la comprensión del sistema.

Separación de preocupaciones: Promueven la reutilización de componentes al modelar sistemas de forma modular, donde cada componente tiene una responsabilidad clara y específica.

Despliegue físico: Aunque son conceptuales, también ayudan a mapear la implementación en artefactos físicos como archivos de código fuente, binarios, o bases de datos.

Ejemplo de un Diagrama de Componentes:

En el proyecto actual se está trabajando en un proyecto para un sistema de validación de transacciones y tickets para la generación de boletos, el cual está dividido en varios componentes:

Cliente Web: Es el componente que proporciona la interfaz de usuario para que los clientes ingresen sus tickets y vean el estado de sus transacciones.

Validación de Tickets: Este componente recibe los datos del cliente, realiza la validación de los tickets (si son válidos o no) y verifica si cumplen con los requisitos.

Base de Datos: Un componente que almacena la información de los tickets y transacciones.

API de Transacciones: Este componente se comunica con la base de datos y el sistema de validación para procesar las transacciones.

Relaciones en este sistema:

El Cliente Web tiene una interfaz que se conecta a la API de Transacciones.

La API de Transacciones proporciona una interfaz a través de la cual el Cliente Web envía las solicitudes.

La Validación de Tickets depende de la Base de Datos para obtener información sobre los tickets y las transacciones pasadas.

Notación en UML:

Cada componente se dibuja como un rectángulo con pequeñas pestañas en la parte superior izquierda.

Las dependencias entre componentes se representan con flechas punteadas.

Las interfaces se representan con pequeños círculos o conectores en el borde de los componentes.

En resumen, los **Diagramas de Componentes** ayudan a visualizar la estructura modular de un sistema, describiendo las relaciones entre los componentes y cómo interactúan a través de sus interfaces, esto es fundamental para planificar la arquitectura de un sistema y garantizar que los componentes sean reutilizables y fácilmente mantenibles, enlace para ver a detalle la simbología de este tipo de diagrama:

<https://drive.google.com/file/d/14ztUTi2EWYZqu8ZdAhbFetxcp-sRWQyO/view?usp=sharing>

Diagrama de Objetos: Muestra instancias específicas de clases en un momento dado.

Diagrama de Despliegue: Representa la distribución física de los componentes en nodos hardware.

Diagrama de Paquetes: Agrupa elementos relacionados para organizar la estructura del sistema.

Diagramas de Comportamiento: Describen cómo los componentes interactúan entre sí y con los usuarios.

Diagrama de Casos de Uso: Define las interacciones entre los usuarios (actores) y el sistema para cumplir objetivos, descrito a detalle dentro de la primera etapa del sistema de gestión del “Sorteo Vacaciones en Familia” en una tienda departamental.

Diagrama de Secuencia: Representa la interacción entre objetos en el tiempo, destacando el orden de los mensajes.

Diagrama de Actividades: Describe el flujo de trabajo o procesos dentro de un sistema.

Elementos Principales de un Diagrama de Actividad:

Actividades (Acciones): Representan una tarea o conjunto de operaciones que se ejecutan. Se dibujan como rectángulos con bordes redondeados.

Flujos de control: Son flechas que indican el orden de ejecución de las actividades.

Decisiones: Representan puntos en los que se debe tomar una decisión, generalmente con múltiples caminos, se dibujan como un rombo.

Inicio y fin:

Nodo inicial: Indica el punto de partida del flujo y se representa como un círculo sólido.

Nodo final: Marca el fin del flujo y se representa como un círculo con un borde grueso.

Bifurcación y Unión (Fork/Join): Permiten modelar paralelismos, se representa con una línea gruesa horizontal o vertical:

Fork: Divide una actividad en varias ramas que se ejecutan en paralelo.

Join: Reúne ramas paralelas en un solo flujo.

Objetos: En algunos casos, se pueden incluir objetos o datos que se manipulan a lo largo del proceso, se muestran como rectángulos conectados a las actividades, enlace para ver a detalle la simbología de este tipo de diagrama:

<https://drive.google.com/file/d/1oA9Nkxbd3iecHGXe6UAVA7uNjb6qJ3Siz/view?usp=sharing>

El modelado UML es una herramienta poderosa que permite diseñar sistemas de software de manera eficiente, proporcionando diagramas que representan tanto la estructura como el comportamiento del sistema, es esencial para el desarrollo colaborativo, la documentación y el análisis de sistemas en diversas etapas del ciclo de vida del software.

Como Jacobson, I. (1992). Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley expresó: “El modelado de casos de uso proporciona una visión clara de cómo los usuarios interactúan con el sistema, ayudando a garantizar que se cumplan tanto los requisitos funcionales como los no funcionales de manera efectiva.”

Descripción:

En el presente documento se realizarán los **Diagramas de Actividades y Componentes** con ayuda de una herramienta de modelado libre o privativa acorde a las actividades anteriores, con ayuda de una herramienta de modelado libre o privativa acorde a las actividades anteriores, que forman parte del **Modelado UML (Unified Modeling Language)**, que es una metodología estandarizada para diseñar y visualizar sistemas de software, proporciona Diagramas que permiten representar la estructura y el comportamiento de un sistema, facilitando la comunicación entre equipos y el Análisis de Requisitos.

Los **Diagramas de Componentes** en UML (Unified Modeling Language) son Diagramas estructurales que representan cómo los componentes de un sistema se organizan y se relacionan entre sí, son particularmente útiles para modelar la arquitectura física de un sistema y cómo los distintos módulos interactúan a nivel de implementación.

Un **Componente** es una parte modular e independiente de un sistema que tiene una funcionalidad bien definida, puede ser un módulo de software, una clase, un subsistema o cualquier parte lógica que contribuya a una solución. Los componentes interactúan entre sí a través de **Interfaces**.

Conceptos Clave en los Diagramas de Componentes:

Componentes: Representan una parte lógica y funcional de un sistema que es independiente y reemplazable, pueden ser módulos de software, bibliotecas, servicios web o cualquier otra parte del sistema que proporcione funcionalidad, se dibujan como rectángulos con dos pestañas en la esquina superior izquierda, lo que simboliza su naturaleza modular.

Ejemplo: En un sistema de validación de transacciones, podrías tener componentes como “Validación de Tickets”, “Procesador de Pagos”, y “Base de Datos”.

Artefactos: Un artefacto es un objeto físico que representa la implementación de un componente, como un archivo ejecutable, un archivo de código fuente, una base de datos o cualquier recurso físico utilizado por el sistema.

En los **Diagramas de Componentes**, un artefacto se dibuja como un rectángulo con la palabra “artefacto” en la esquina superior izquierda.

Ejemplo: Un componente lógico “Base de Datos” podría estar vinculado a un artefacto físico que representa el archivo de base de datos o el sistema de gestión de bases de datos utilizado (como un archivo .db o una instancia de SQL Server).

Interfaces: Las interfaces definen el contrato entre dos componentes, son los puntos de contacto o interacción que permiten a los componentes comunicarse entre sí, se dibujan como pequeños círculos adheridos a los bordes de los componentes, pueden ser:

Interfaces proporcionadas: Lo que un componente ofrece, se representan con un círculo simple (similar a un puerto).

Interfaces requeridas: Lo que un componente necesita para funcionar, se representa con un semicírculo o “socket” en el borde del componente.

Ejemplo: Un componente “API de Validación” puede proporcionar una interfaz para que el “Cliente Web” envíe solicitudes de validación de tickets, el componente “Validación de Tickets” podría requerir una interfaz proporcionada por el componente “Base de Datos” para obtener información sobre los tickets.

Relaciones entre Componentes:

Dependencias: Una relación de dependencia indica que un componente necesita otro para funcionar correctamente, se representan con una línea punteada con una flecha desde el componente dependiente hacia el componente del que depende.

Relaciones de ensamblaje (Assembly): Un ensamblaje conecta una interfaz proporcionada con una interfaz requerida, mostrando que dos componentes están directamente conectados y colaboran entre sí.

Relaciones de delegación: En casos más complejos, un componente puede delegar parte de su trabajo a otro, lo que se indica con una línea de delegación.

Puertos: Los puertos representan puntos donde un componente interactúa con el mundo exterior o con otros componentes, a menudo se usan cuando un componente tiene múltiples interfaces o puntos de interacción, representado por un pequeño cuadrado en el borde del componente, y puede conectarse a varias interfaces.

Elementos Gráficos en un Diagrama de Componentes:

Componentes: Representados como rectángulos con dos pestañas en la parte superior izquierda.

Artefactos: Son rectángulos con la etiqueta “Artefacto” en la parte superior izquierda, vinculados a componentes.

Interfaces: Pequeños círculos que representan los puntos de conexión de un componente.

Interfaces proporcionadas: Representadas como un círculo que sale del componente.

Interfaces requeridas: Representadas como un semicírculo o “Socket”.

Relaciones: Las dependencias se muestran con líneas punteadas y las relaciones de ensamblaje con líneas sólidas que conectan las interfaces.

Ejemplo Práctico de un Diagrama de Componentes:

En esta actividad se está trabajando en un sistema para la gestión y validación de tickets, con los **siguientes componentes:**

Cliente Web:

Proporciona la interfaz de usuario para que los clientes puedan ingresar sus tickets y verificar el estado de su transacción, tiene una interfaz proporcionada para enviar solicitudes a la API de validación.

API de Validación de Tickets:

Recibe solicitudes del Cliente Web y realiza la validación de los tickets ingresados, depende de un componente Base de Datos para acceder a la información de los tickets, proporciona una interfaz a la cual el Cliente Web puede conectarse.

Base de Datos:

Almacena la información sobre los tickets y transacciones previas, proporciona una interfaz para que otros componentes (como la API de Validación de Tickets) puedan acceder a los datos almacenados.

Servicio de Notificación:

Envía notificaciones a los clientes una vez que la validación del ticket ha sido completada.

Los **Diagramas de Componentes** permiten modelar de manera clara y precisa la arquitectura física y lógica de un sistema de software, al descomponer el sistema en componentes reutilizables, estos diagramas no solo facilitan la implementación y el mantenimiento, sino que también promueven la escalabilidad y la interoperabilidad, representan el esqueleto sobre el cual se construye una arquitectura de software robusta, facilitando tanto la comunicación entre los equipos de desarrollo como la planificación eficiente del despliegue y mantenimiento del sistema.

Por su parte, los **Diagramas de Actividad** son un tipo de diagrama de comportamiento en **UML (Unified Modeling Language)** que representan el flujo de control o el proceso de una

actividad o conjunto de actividades dentro de un sistema, son útiles para modelar flujos de trabajo, procesos de negocio, o cualquier proceso que implique la secuencia de acciones y decisiones.

Sus Elementos Son:

Actividades o Acciones:

Representan un paso o tarea dentro del proceso.

Se dibujan como rectángulos con esquinas redondeadas.

Pueden involucrar operaciones simples (como una tarea única) o complejas (como subprocesos).

Ejemplo: “Validar pago”, “Enviar notificación”, “Seleccionar ticket”.

Flujos de Control (Transiciones):

Son las flechas que conectan las actividades, indicando la secuencia de ejecución.

Pueden ser flujos secuenciales o alternativos (en función de decisiones o condiciones).

Ejemplo: Tras “Validar pago”, el flujo continúa hacia “Generar factura”.

Decisiones y Condiciones:

Se utilizan para modelar puntos en los que una decisión debe tomarse.

Se representa con un rombo, y las posibles opciones (rutas) salen de él.

Cada ruta tiene una condición que debe cumplirse para que el flujo siga por ese camino.

Ejemplo: Después de “Revisar saldo”, una decisión podría bifurcar el flujo en “Saldo suficiente” o “Saldo insuficiente”.

Fork (Bifurcación) y Join (Unión):

Fork (Bifurcación): Divide una actividad en varias acciones paralelas, que pueden ejecutarse simultáneamente.

Join (Unión): Reúne varias acciones paralelas en una sola línea de flujo.

Ejemplo: Después de “Seleccionar tickets”, el proceso podría bifurcarse en “Verificar ticket 1” y “Verificar ticket 2”, que pueden ejecutarse en paralelo, una vez verificados, las acciones se pueden unir antes de continuar.

Nodo Inicial y Nodo Final:

Nodo inicial: Se representa como un pequeño círculo negro y marca el comienzo del flujo de actividades.

Nodo final: Se representa como un círculo con un contorno grueso y negro, y marca el fin del proceso o actividad.

Ejemplo: El nodo inicial podría representar “Iniciar sesión en el sistema”, y el nodo final sería “Confirmación enviada”.

Regiones Concurrentes:

Permiten dividir un proceso en múltiples actividades que ocurren simultáneamente dentro de una misma región del diagrama.

Cada región maneja un conjunto independiente de actividades.

Ejemplo: Mientras una parte del proceso maneja la “Validación de usuario”, otra puede estar gestionando el “Registro de transacciones”.

Swimlanes (Carriles):

Usados para organizar actividades según las responsabilidades de diferentes actores o sistemas, cada **swimlane** representa un participante (actor, sistema, módulo) y contiene las actividades que esa entidad ejecuta.

Ejemplo: En un proceso de compra, podrías tener carriles para el cliente, el sistema de pagos, y el sistema de envíos.

Objetos: Son representaciones de datos o artefactos que son manipulados por las actividades, suelen estar conectados a actividades con líneas punteadas, mostrando que la actividad usa o genera ese objeto.

Ejemplo: Un objeto podría ser un “Ticket” que se procesa o un “Recibo” que se genera tras una compra.

Flujo de Objetos:

Representa cómo los datos u objetos fluyen entre las actividades, las líneas punteadas conectan las actividades con los objetos.

Ejemplo: La acción “Generar ticket” puede estar conectada con un objeto “Ticket” que luego se entrega a la acción “Verificar ticket”.

Ejemplo en un Proyecto de Validación de Transacciones:

Inicio del Proceso: Un cliente inicia la validación de transacciones.

Acción 1: Verificación de la autenticidad de los tickets.

Decisión: Si los tickets son válidos, el flujo continúa; de lo contrario, se rechaza.

Acciones Concurrentes: Mientras se verifican los tickets, también se comprueba si los fondos son suficientes en la cuenta del cliente.

Unión: Una vez completadas ambas tareas, el sistema decide si se envía una confirmación de la transacción.

Fin del Proceso: Se genera una notificación para el cliente indicando el estado de la transacción.

Este tipo de Diagramas es esencial cuando se requiere modelar secuencias complejas de acciones o procesos con múltiples decisiones y paralelismos.

Justificación:

El objetivo de este proyecto es subrayar la relevancia del uso del **Lenguaje Unificado de Modelado (UML)** como una herramienta clave en el desarrollo de sistemas de software, promoviendo un enfoque estructurado y eficiente, en este contexto, se implementará un sistema integral para la gestión del “**Sorteo Vacaciones en Familia**”, el cual estará vinculado a compras y transacciones en una tienda departamental, la necesidad de un diseño nítido y bien estructurado es imperativa para asegurar que tanto desarrolladores como partes interesadas comprendan plenamente los procesos y flujos de trabajo implicados en el sistema.

El uso de UML es crucial en este sentido por diversas razones:

Claridad en la Comunicación: UML facilita la creación de una representación estandarizada de la arquitectura del sistema, lo que optimiza la comunicación entre los distintos actores involucrados (desarrolladores, analistas y gerentes), asegurando una visión compartida y coherente de los objetivos y el diseño del sistema.

Eficiencia en la Organización: Mediante la utilización de diagramas de actividades y componentes, UML permite modelar procesos complejos de manera ordenada, favoreciendo la planificación y ejecución del sistema. Esto resulta esencial para que funciones críticas como la validación de transacciones, la emisión de boletos y la participación en el sorteo operen de forma fluida y cohesionada.

Control y Transparencia en la Gestión: El sistema proporcionará a administradores y empleados las herramientas necesarias para gestionar parámetros fundamentales del sorteo, tales como la verificación de ganadores y la asignación de boletos, asegurando una operación eficaz y transparente.

En esta fase del proyecto, el enfoque recae sobre la creación de diagramas de actividades y componentes UML para modelar de manera precisa tanto el flujo de los procesos como las interacciones entre los diversos componentes del sistema.

Esta etapa es crítica por las siguientes razones:

Modelado del Comportamiento del Sistema: Los diagramas de actividades permiten una representación clara de los procesos centrales, tales como la verificación de transacciones, la generación de boletos y la validación de términos y condiciones, esta visualización es indispensable para asegurar que cada etapa del flujo de trabajo esté correctamente definida y que las reglas del sistema sean respetadas.

Descomposición y Representación Arquitectónica: A través de los diagramas de componentes, se desglosa el sistema en módulos independientes (como el sistema de transacciones, el sistema bancario, la gestión de boletos, entre otros), facilitando la identificación de relaciones y dependencias, esto contribuye a una arquitectura más clara, mantenible y adaptable.

Escalabilidad y Reutilización: La separación modular del sistema fomenta la reutilización de componentes en futuras expansiones, lo que favorece la escalabilidad y flexibilidad del proyecto a largo plazo.

Validación y Control de Calidad: El modelado de procesos y componentes no solo garantiza el cumplimiento de los requisitos definidos, sino que también minimiza la posibilidad de errores, actuando como una herramienta de validación previa a la implementación de fases más avanzadas del desarrollo.

En resumen, esta etapa del proyecto es esencial para generar una representación visual precisa del flujo de trabajo y la estructura del sistema, asegurando que todos los componentes

operen de manera armónica y que el sistema sea escalable, mantenible y eficiente.

Los diagramas de componentes permiten visualizar la interacción entre las distintas partes del sistema, como el cliente, el sistema bancario y el sistema web, mientras que los diagramas de actividades detallan los flujos de trabajo involucrados, como la validación de transacciones y la gestión de boletos para el sorteo

Desarrollo

Contextualización:

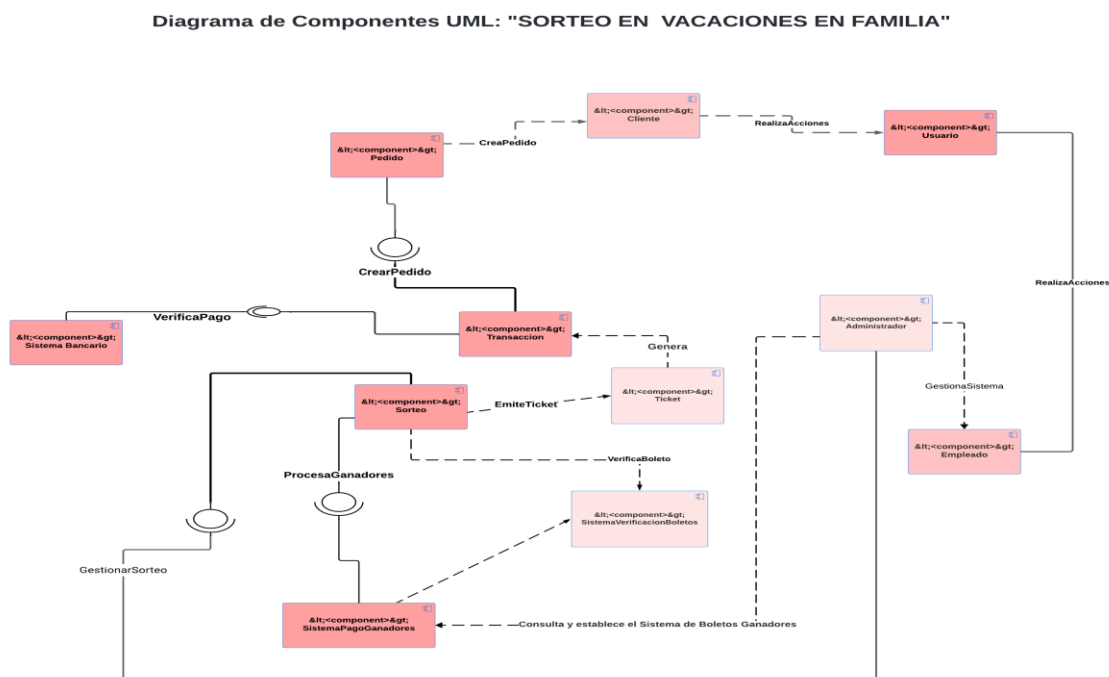
Modelar los Diagramas permite identificar elementos que no son vistos de manera simple, y encontrar el vínculo entre los mismos; por tal motivo a continuación se pretende reconocer la importancia de los Diagramas de actividad y componentes, los cuales muestran cómo los eventos de un caso de uso se relacionan o como se pueden coordinar para el flujo de un sistema y sus funcionalidades.

Actividad:

Diseñar los **Diagramas de Actividades y Componentes** con ayuda de una herramienta de modelado libre o privativa acorde a las actividades anteriores.

Figura 1

Diagrama de Componentes UML: “SORTEO EN VACACIONES EN FAMILIA”:



Nota: La imagen es un **Diagrama de Componentes UML** titulado “Sorteo en Vacaciones en Familia”, que representa la interacción entre los diferentes componentes del sistema que gestionan un sorteo vinculado a compras y transacciones:

Componentes del sistema:

Cliente: Este componente representa al cliente que realiza pedidos, el cliente interactúa con el sistema creando pedidos y realizando acciones como verificar el estado de los boletos emitidos para el sorteo.

Usuario: Este componente agrupa las acciones generales de cualquier tipo de usuario (Clientes, Empleados, Administradores). Se conecta con el componente Cliente y el componente Empleado.

Pedido: El pedido es creado por el cliente, se muestra el flujo donde un cliente puede crear un pedido, que luego se conecta a las transacciones para verificar el pago.

Transacción: Este componente maneja las transacciones de pago, una vez que se crea un pedido, este pasa por el sistema de transacciones, que a su vez verifica el pago con el sistema bancario.

Sistema Bancario: Este componente verifica el pago del pedido, si el pago es exitoso, el sistema permite generar un ticket y proceder con el sorteo.

Sorteo: Una vez que el pago ha sido verificado, el sistema emite boletos y los asigna al sorteo correspondiente, el componente sorteo está conectado a la emisión de tickets y a la verificación de boletos.

Ticket: Este componente emite un ticket al cliente una vez que la transacción ha sido procesada, está relacionado con el sorteo y la verificación de boletos.

Sistema de Verificación de Boletos: Este componente verifica los boletos emitidos para

el sorteo, se conecta con el componente de sorteo y el sistema de pago de ganadores para verificar cuáles son los boletos ganadores.

Sistema Pago Ganadores: Procesa el pago a los ganadores del sorteo, una vez que los boletos ganadores han sido verificados.

Administrador: El Administrador gestiona el sistema, estableciendo los parámetros del sorteo, verificando que todo esté funcionando correctamente, y emitiendo notificaciones en caso de que haya algún problema.

Empleado: El empleado también interactúa con el sistema, gestionando las actividades relacionadas con el sorteo, como la verificación de ganadores o la gestión de boletos emitidos.

Flujos de proceso:

El cliente **crea un pedido**, que luego es procesado a través del **sistema de transacciones**.

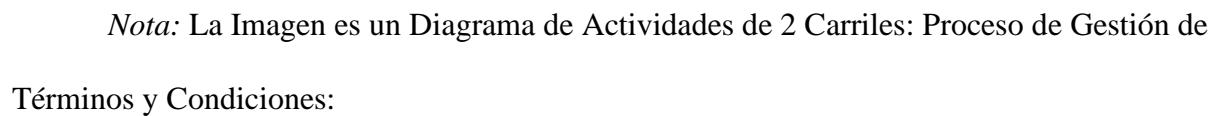
El **sistema bancario** se encarga de **verificar el pago**. Si es aprobado, se emite un **ticket** y el cliente es ingresado en el **sorteo**.

El sistema **procesa los ganadores y verifica los boletos** ganadores, asegurando que el sistema de sorteos emita correctamente los resultados.

Finalmente, el **sistema de pago de ganadores** distribuye los premios a los clientes con boletos ganadores.

Este diagrama describe cómo se gestiona el proceso completo del sorteo de “Vacaciones en Familia”, desde la creación del pedido, la verificación del pago, la emisión de boletos, la participación en el sorteo y el procesamiento de los ganadores, el flujo incluye tanto las interacciones entre el cliente y el sistema, como el rol de los Administradores y empleados en la gestión y verificación del sorteo, para mas detalles consulte el siguiente enlace:

Diagrama de Actividades 1: Establecimiento Condiciones y Términos:



Carril: Administrador del Sistema

Este carril refleja todas las actividades y decisiones que lleva a cabo el Administrador del Sistema, quien tiene la responsabilidad de definir, revisar, corregir y aprobar los Términos y Condiciones dentro del sistema; Las actividades detalladas en este carril cubren desde el inicio del proceso hasta la aprobación final, garantizando que los términos sean claros, precisos y legalmente válidos para ser presentados a los usuarios.

Iniciar el establecimiento de Términos y Condiciones (Actividad):

Descripción: El Administrador del Sistema inicia el proceso de creación o modificación de los términos y condiciones. Esta actividad marca el inicio del flujo de trabajo para establecer las reglas que regirán el sistema.

Significado: Este paso es esencial, ya que da comienzo a todo el proceso, iniciar el establecimiento de los términos implica que el Administrador asume el control del proceso y es responsable de desencadenar el flujo de actividades que siguen, desde la definición inicial hasta la aprobación y validación final de los términos, sin esta actividad, el sistema no puede proceder a realizar las validaciones y revisiones necesarias.

Definir Términos y Condiciones (Actividad):

Descripción: En esta actividad, el Administrador redacta o establece los Términos y Condiciones que los usuarios del sistema deberán aceptar para poder utilizarlo, esto incluye la creación de las reglas, las políticas y las normativas que definirán el uso adecuado del sistema.

Significado: Esta es una de las actividades más importantes del Administrador, ya que implica la creación de las disposiciones que regirán el comportamiento de los usuarios dentro del sistema; Los Términos y Condiciones pueden abarcar una amplia variedad de temas, como las obligaciones de los usuarios, las garantías ofrecidas por el sistema, las políticas de privacidad y

los derechos y responsabilidades tanto de los Administradores como de los usuarios. La precisión y claridad en la redacción de estos términos es crucial para evitar malentendidos futuros y asegurar el cumplimiento de normativas legales y operativas.

Corregir inconsistencias (Actividad):

Descripción: Si durante el proceso de validación o revisión de los Términos y Condiciones se detectan errores o incongruencias, el Administrador del Sistema es el encargado de corregir dichas inconsistencias, esta actividad puede ser iniciada de manera automática por el sistema si los términos no cumplen con los criterios establecidos, o puede ser resultado de una revisión manual.

Significado: La corrección de inconsistencias es fundamental para asegurar que los Términos y Condiciones sean claros, precisos y completos, las inconsistencias pueden surgir de errores tipográficos, omisiones de apartados importantes o incoherencias en la redacción, el sistema no puede proceder con la aprobación o publicación de términos que contengan errores, por lo que este paso es obligatorio antes de que el flujo continúe, es una medida de control que garantiza la calidad del contenido antes de su implementación.

Aprobar Términos y Condiciones (Actividad):

Descripción: Una vez que los Términos y Condiciones han sido revisados, corregidos y validados, el Administrador del Sistema procede a aprobarlos formalmente, esta aprobación significa que los términos están listos para ser implementados en el sistema y que los usuarios deberán aceptarlos para poder continuar usando la plataforma.

Significado: La aprobación final marca el fin del proceso de creación y revisión de los términos y condiciones, a partir de este punto, los términos están listos para ser aplicados y presentados a los usuarios, este es un paso crucial, ya que una vez aprobados, los términos pasan

a ser de cumplimiento obligatorio para los usuarios del sistema. El Administrador del Sistema asume la responsabilidad de garantizar que los términos aprobados sean claros, completos y estén alineados con los objetivos y las normativas legales aplicables.

Carril: Sistema de Gestión de Términos y Condiciones

Este carril refleja las actividades automatizadas o semi-automatizadas que realiza el sistema encargado de gestionar los Términos y Condiciones una vez que el Administrador los ha definido o corregido. El sistema es responsable de garantizar que los términos se almacenen correctamente, que se notifique a los usuarios sobre cualquier cambio y que los términos cumplan con ciertos criterios preestablecidos antes de ser presentados para su aprobación final.

Guardar Términos y Condiciones (Actividad):

Descripción: Una vez que el Administrador del Sistema ha definido los términos y condiciones, el sistema se encarga de almacenarlos de forma segura en su base de datos o en un repositorio designado, este paso asegura que la versión más reciente de los términos esté disponible para futuras consultas o revisiones.

Significado: Guardar los Términos y Condiciones en un repositorio centralizado es crucial para mantener un registro de versiones y garantizar que los términos estén accesibles para revisiones o auditorías futuras, este almacenamiento también permite al sistema gestionar actualizaciones y garantizar que todos los usuarios tengan acceso a la versión más reciente de los términos.

Generar Notificación de Términos y Condiciones (Actividad):

Descripción: Después de que los términos han sido guardados en el sistema, este genera una notificación automatizada para informar a los usuarios o a otros actores relevantes del sistema (como gerentes o auditores) que los términos han sido creados o actualizados.

Significado: La notificación asegura que todos los involucrados estén al tanto de cualquier cambio en los términos y condiciones, esta notificación puede ser presentada a los usuarios en su próxima sesión o enviada por correo electrónico, dependiendo de la configuración del sistema, es un paso importante para garantizar la transparencia y mantener a los usuarios informados de los términos que rigen su uso del sistema.

Validar Términos y Condiciones (Actividad):

Descripción: El sistema realiza una validación automática de los términos y condiciones, asegurando que cumplan con ciertos criterios de calidad y formato, esto incluye verificar que todos los apartados requeridos estén presentes, que no haya campos vacíos, errores tipográficos o incoherencias en la redacción.

Significado: La validación automática es un paso crucial que permite al sistema identificar errores antes de que los términos sean presentados para su aprobación final, al asegurar que los términos cumplen con los requisitos establecidos, se reduce la posibilidad de errores humanos y se garantiza que el contenido sea de alta calidad. Si la validación falla, los términos se devuelven al Administrador del Sistema para que realice las correcciones necesarias.

Decisión: ¿Los Términos son válidos? (Decisión):

Descripción: En este punto, el sistema toma una decisión basada en los resultados de la validación. Si los términos cumplen con todos los criterios de validación, el flujo de actividades continúa hacia la revisión final o aprobación por parte del Administrador. Si no cumplen con los criterios, los términos se devuelven al Administrador para que realice las correcciones necesarias.

Resultados posibles:

Sí: Los términos son válidos y el flujo continúa hacia su revisión y aprobación final.

No: Si los términos no son válidos, el flujo regresa al Administrador del Sistema, quien debe corregir las inconsistencias detectadas antes de que el proceso pueda continuar.

Significado: Este paso de decisión es fundamental para garantizar que solo los términos correctamente validados avancen en el flujo de trabajo. Si los términos contienen errores o no cumplen con los estándares, el sistema evita que sean aprobados o publicados hasta que hayan sido corregidos, esto asegura que los usuarios nunca vean términos incompletos o erróneos.

Ejemplo de Flujo Detallado:

El Administrador del Sistema inicia el proceso al definir los términos y condiciones. Una vez definidos, el sistema los guarda en su repositorio y genera una notificación para informar a los usuarios.

El sistema luego valida los términos para asegurarse de que cumplan con todos los criterios establecidos. Si los términos son válidos, el proceso continúa hacia la revisión o aprobación.

Si la validación falla, los términos se devuelven al Administrador, quien deberá corregir cualquier inconsistencia antes de que el flujo pueda avanzar.

Descripción Completa del Flujo por Carriles:

Administrador del Sistema:

Inicia el proceso de definición de términos y condiciones.

Si el sistema detecta errores o inconsistencias, el Administrador los corrige.

Una vez corregidos y validados, el Administrador aprueba los términos, completando así el proceso.

Sistema de Gestión de Términos y Condiciones:

Guarda los Términos y Condiciones en su base de datos.

Genera una notificación para informar a los usuarios y otros actores del sistema sobre la creación o actualización de los términos.

Valida que los términos cumplan con los requisitos establecidos.

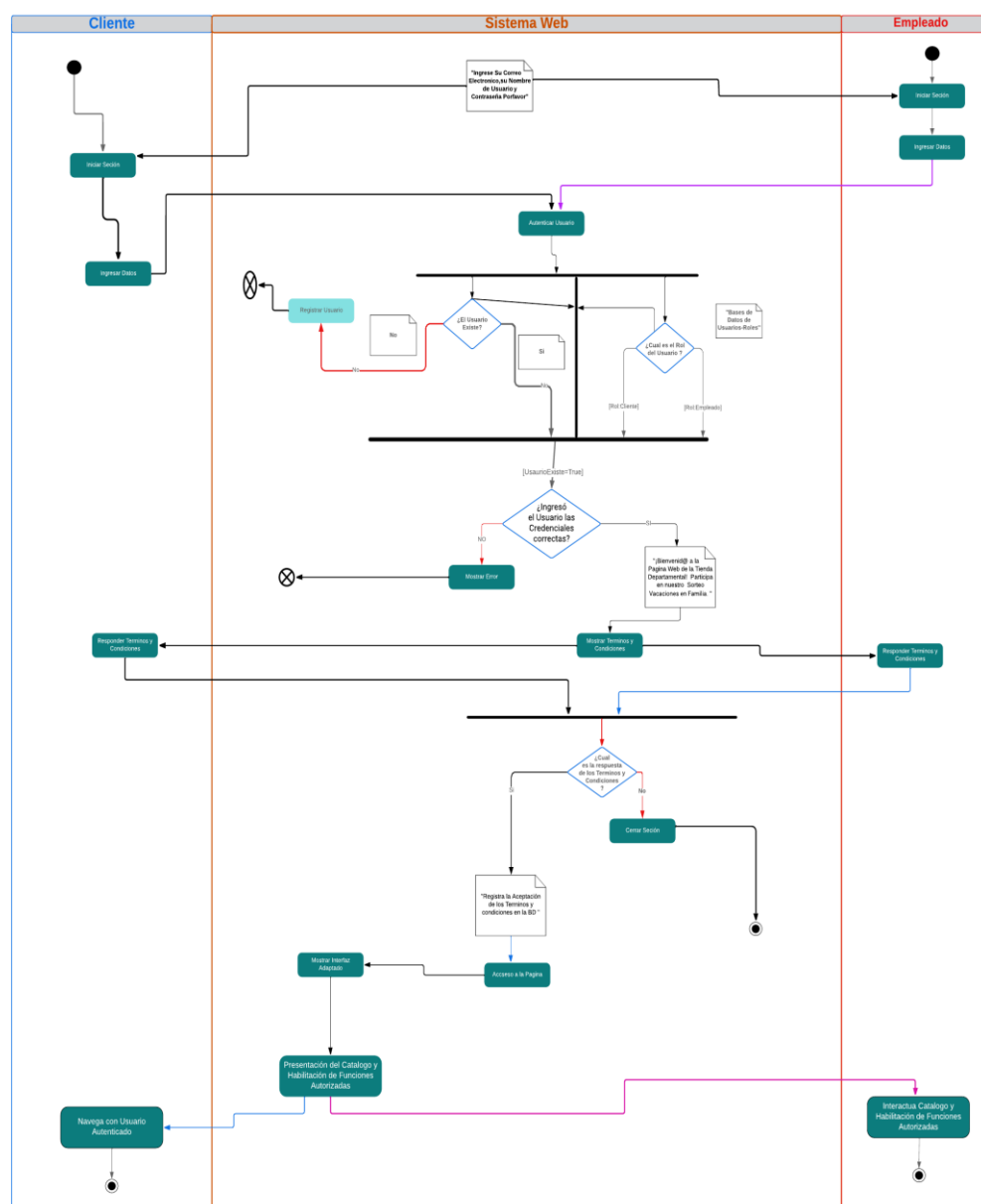
Si los términos son válidos, el sistema permite que el proceso continúe. Si no lo son, los devuelve al Administrador para correcciones.

El sistema automatiza gran parte del proceso de validación y notificación, mientras que el Administrador del Sistema asume la responsabilidad de definir, revisar y corregir los términos cuando sea necesario, ambos carriles interactúan constantemente para garantizar que los Términos y Condiciones sean precisos y estén disponibles para su implementación sin errores, para más detalles consulte el siguiente enlace: https://lucid.app/lucidchart/887f744d-1d51-40b2-a32e-5ed82d2c1a81/edit?viewport_loc=-5564%2C-3182%2C4440%2C2135%2C0_0&invitationId=inv_d1e59a50-4f53-417f-9b29-3bcdf8c7b75c

Figura 3

Diagrama de Actividades 2 con carriles UML: Verificación de Inicio de Sesión

Diagrama de Actividades con Carriles UML: Verificación de Inicio de Sesión



Nota: La imagen es un Diagrama de Actividades de Carriles que se centra en el proceso

de verificación de inicio de sesión, dividido en varios carriles para representar diferentes actores (Cliente, Sistema Web, Empleado). **Descripción detallada del flujo:**

Carril 1: Cliente

Este carril refleja las acciones realizadas por el cliente al interactuar con el sistema web para iniciar sesión.

Iniciar Sesión (Actividad)

Descripción: El cliente ingresa al sistema e intenta iniciar sesión, esta actividad marca el comienzo del flujo de interacción entre el cliente y el sistema.

Significado: El intento de iniciar sesión es el punto de entrada para acceder a las funciones protegidas del sistema, es el primer paso que debe realizar el usuario para ser autenticado y utilizar las funcionalidades que ofrece el sitio web.

Ingresar Datos (Actividad)

Descripción: El cliente introduce sus credenciales de inicio de sesión, como su nombre de usuario y contraseña.

Significado: Esta etapa es crucial para la autenticación. El sistema necesita los datos del cliente para proceder a validarlos en los siguientes pasos del flujo.

Recibir Mensaje de Error (Decisión)

Descripción: Si la validación de las credenciales falla, el cliente recibe un mensaje de error indicando que el inicio de sesión no fue exitoso.

Significado: Esto asegura que el cliente sea consciente de que hubo un problema con las credenciales ingresadas, y puede intentar nuevamente o seguir el procedimiento para la recuperación de la cuenta.

Navegar con Usuario Autenticado (Actividad)

Descripción: En caso de que el cliente haya ingresado las credenciales correctas y haya sido autenticado, se le permite navegar por las secciones del sistema web que están habilitadas para usuarios autenticados.

Significado: Aquí culmina el flujo para un cliente con éxito en su intento de iniciar sesión, dándole acceso a las funciones disponibles solo para usuarios registrados.

Carril 2: Sistema Web

El sistema web tiene un rol central, ejecutando la mayoría de las acciones automatizadas y validaciones que aseguran que el flujo de autenticación se realice correctamente.

Autenticar Usuario (Actividad)

Descripción: El sistema toma las credenciales ingresadas por el cliente y verifica su validez contra la base de datos de usuarios registrados.

Significado: Este paso es crítico para la seguridad del sistema, ya que el sistema decide si las credenciales son correctas y si el usuario puede o no acceder a las funcionalidades protegidas.

Verificar Términos y Condiciones (Actividad)

Descripción: El sistema comprueba si el cliente ha aceptado los términos y condiciones actualizados del servicio.

Significado: Este paso es importante para asegurar que los usuarios estén de acuerdo con los términos más recientes, un aspecto clave tanto desde un punto de vista legal como de cumplimiento de normas internas.

Mostrar Error (Decisión)

Descripción: Si el cliente no ha aceptado los términos y condiciones, el sistema lo redirige para que acepte los términos antes de proceder.

Significado: Este paso garantiza que ningún cliente pueda acceder al sistema sin aceptar

los términos y condiciones, asegurando que los usuarios cumplan con las políticas del servicio.

Validar Credenciales (Actividad)

Descripción: El sistema valida las credenciales proporcionadas por el cliente, comparándolas con los registros en la base de datos.

Significado: Este es el paso donde se define si las credenciales son correctas. Si lo son, el sistema permite al cliente continuar. Si no, se devuelve un error.

Generar Token de Sesión (Actividad)

Descripción: Una vez que las credenciales son validadas, el sistema genera un token de sesión que representa la autenticación del cliente para futuras interacciones dentro del sistema.

Significado: Este paso asegura que el cliente pueda realizar acciones adicionales sin tener que volver a autenticarse en cada solicitud, y proporciona una capa adicional de seguridad al sistema.

Presentación del Catálogo y Funciones Autorizadas (Actividad)

Descripción: Si el cliente pasa todas las validaciones, el sistema le muestra el catálogo o las funciones autorizadas dentro del sistema web.

Significado: Este es el destino final del flujo de autenticación. El cliente ahora tiene acceso a todas las funciones que su rol dentro del sistema le permite utilizar.

Carril 3: Empleado

Este carril refleja las actividades que realiza un empleado, que podría ser un actor distinto al cliente dentro del flujo del sistema.

Iniciar Sesión (Actividad)

Descripción: El empleado, al igual que el cliente, intenta iniciar sesión ingresando sus credenciales.

Significado: Este es el primer paso para que un empleado acceda al sistema y utilice sus funcionalidades especializadas.

Ingresar Datos (Actividad)

Descripción: El empleado introduce su nombre de usuario y contraseña en el sistema.

Significado: Este paso es equivalente al realizado por el cliente, pero en este caso el sistema puede otorgarles acceso a diferentes funciones basadas en su rol.

Recibir Mensaje de Error (Decisión)

Descripción: Si las credenciales ingresadas no son correctas, el empleado recibe un mensaje de error y no puede proceder.

Significado: El sistema asegura que los empleados solo puedan acceder si sus credenciales son válidas.

Interacción con Catálogo y Funciones Habilitadas para Empleados (Actividad)

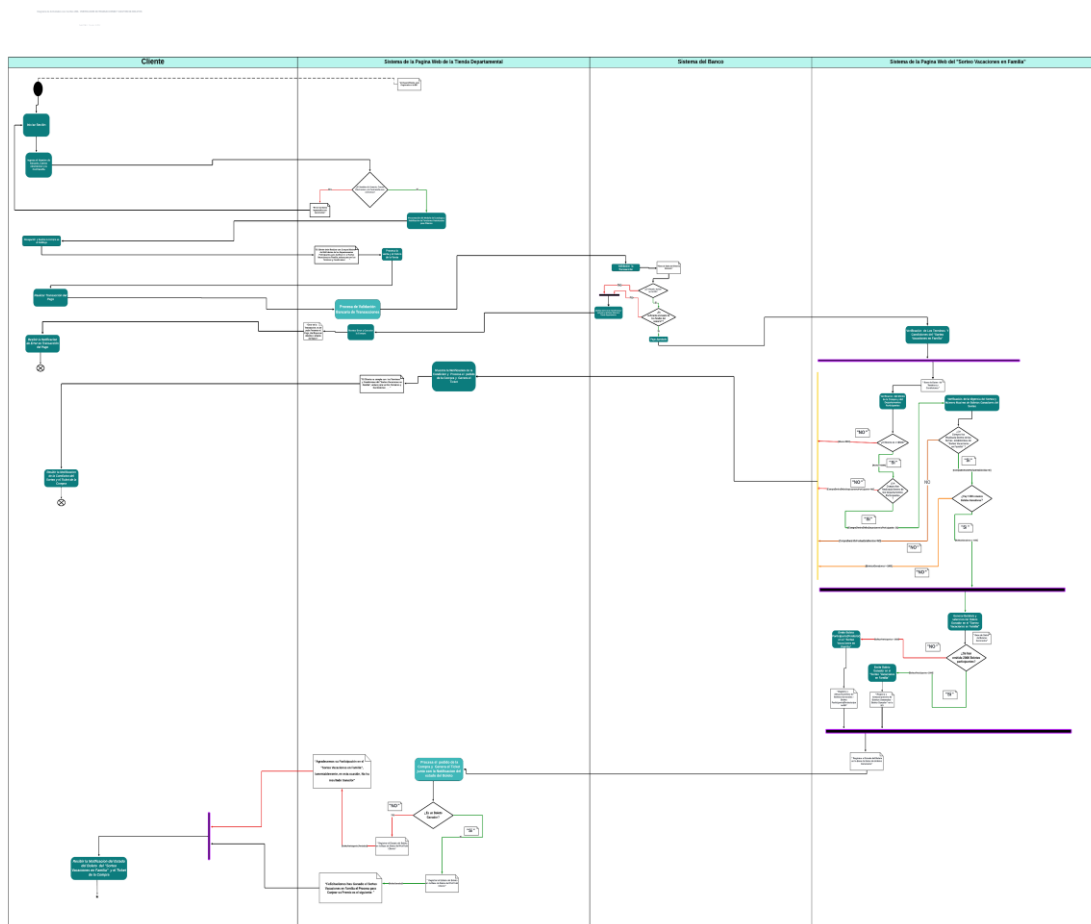
Descripción: Si el empleado logra autenticarse, puede acceder a las funciones que están autorizadas para su rol.

Significado: Este es el paso final para los empleados, permitiéndoles acceder a las áreas del sistema relevantes para su trabajo.

Este flujo cubre todas las etapas críticas de la autenticación de usuarios y empleados dentro de un sistema, asegurando que solo las personas con las credenciales correctas puedan acceder y usar las funciones que les corresponden, para más detalles consulte el siguiente enlace:
https://lucid.app/lucidchart/82cb0d05-3952-4ac9-9999-ec9d8c22b471/edit?viewport_loc=699%2C1297%2C1698%2C817%2C0_0&invitationId=inv_b157749a-fe6a-452f-b814-af8b02910d25

Figura 4

Diagrama de Actividades 3 con Carriles UML: VERIFICACION DE TRANSACCIONES Y GESTION DE BOLETOS



Nota: La imagen es un diagrama de actividades con carriles UML enfocado en la **Verificación de Transacciones y Gestión de Boletos.**

Descripción detallada de las secciones y actividades representadas en el diagrama:

Carril 1: Cliente

Este carril representa las acciones que el cliente realiza dentro del sistema.

Iniciar Sesión (Actividad)

El cliente inicia su sesión en el sistema de la tienda departamental, introduciendo sus credenciales.

Realizar una Transacción Bancaria (Actividad)

El cliente intenta realizar una transacción bancaria para, por ejemplo, la compra de boletos para el sorteo.

Recepción del Resultado de la Transacción (Decisión)

Tras la transacción bancaria, el cliente recibe una notificación del sistema informando si la transacción fue exitosa o si falló. En caso de éxito, continúa con el proceso.

Generar Boletos de Participación (Actividad)

Si la transacción es exitosa, el cliente obtiene boletos de participación en el sorteo de vacaciones en familia, de lo contrario, se le notifica del error y el proceso se detiene.

Carril 2: Sistema de la Página Web de la Tienda Departamental

En este carril, se muestran las actividades realizadas automáticamente por el sistema de la tienda departamental.

Verificación de Credenciales (Actividad)

El sistema valida las credenciales del cliente cuando este intenta iniciar sesión.

Validación de Transacciones (Actividad)

Una vez realizada la transacción bancaria, el sistema verifica que esta se haya completado correctamente.

Proceso de Validación de Términos y Condiciones (Decisión)

El sistema valida si el cliente ha aceptado los términos y condiciones, Si no los ha aceptado, se le redirige para que lo haga.

Generación de Boletos (Actividad)

Si la transacción es válida y el cliente ha aceptado los términos, se generan los boletos de participación para el sorteo.

Notificación de Transacción Exitosa (Actividad)

El sistema notifica al cliente que la transacción ha sido exitosa y que los boletos han sido generados correctamente.

Carril 3: Sistema del Banco

Este carril ilustra las actividades que corresponden al sistema bancario.

Recepción de Solicitud de Transacción (Actividad)

El sistema del banco recibe la solicitud de transacción del cliente desde la página web de la tienda departamental.

Validación de Fondos Disponibles (Decisión)

El sistema verifica si el cliente tiene los fondos suficientes para completar la transacción. Si no, la transacción falla.

Ejecución de la Transacción (Actividad)

Si hay fondos suficientes, el sistema procede con la ejecución de la transacción y la notificación del resultado.

Carril 4: Sistema de la Página Web del Sorteo “Vacaciones en Familia”

Este carril muestra el flujo de actividades que se gestionan en el sistema del sorteo en sí.

Validación de la Vigencia del Sorteo (Actividad)

El sistema verifica si el sorteo aún está dentro de su periodo de vigencia.

Validación de Boletos Disponibles (Decisión)

Se comprueba si aún existen boletos ganadores disponibles. Si no quedan boletos ganadores, se notifica que el sorteo ha concluido.

Gestión de Boletos Ganadores (Actividad)

En el caso de que la transacción sea válida y existan boletos ganadores, el sistema gestiona la generación de boletos para los clientes ganadores y actualiza el conteo.

Generación del Informe Final (Actividad)

Una vez alcanzado el límite de boletos ganadores o concluido el plazo de vigencia del sorteo, el sistema genera un informe final para los Administradores.

Conclusión del Flujo:

El diagrama finaliza con la posibilidad de que el sorteo termine si se alcanzan los boletos ganadores o si el plazo finaliza. En ambos casos, se genera un informe final que concluye el proceso de verificación y gestión de boletos.

El flujo ilustra cómo las interacciones entre el cliente, el sistema de la tienda, el sistema del banco y el sistema del sorteo están interconectadas para asegurar que solo los clientes que cumplan con los requisitos puedan participar en el “**Sorteo de Vacaciones en Familia**”.

La correcta validación de cada paso asegura la integridad del sistema y el cumplimiento de las reglas del sorteo, para mayores detalles seguir el siguiente enlace:

https://lucid.app/lucidchart/c47c8ec1-ad65-4f88-b8fa-41f9701c8820/edit?view_items=CkX.ADhFuwXS&invitationId=inv_9b5c94fc-957d-4064-8516-14e3edc3e04b

Conclusión:

La Adquisición del conocimiento de **UML (Unified Modeling Language)**, específicamente de los **Diagramas de Actividades y Componentes**, tiene una gran relevancia tanto en la vida laboral como en situaciones cotidianas, particularmente en los ámbitos de la ingeniería de software, la gestión de proyectos y la resolución de problemas.

En la **Vida Laboral** el **Diagramas de Actividades** permite claridad en los procesos, visualizar los flujos de trabajo y los procesos secuenciales, facilitando la comprensión de cómo se ejecutan las tareas dentro de un sistema, esto es clave en empresas donde es necesario modelar flujos complejos, como en la gestión de transacciones o la automatización de procesos.

El **Diagrama de Componentes** ayuda al **Diseño Eficiente de Sistemas**, estructurar un sistema mostrando sus módulos y sus relaciones, en el Desarrollo de Software, este tipo de diagrama es esencial para identificar los elementos de la arquitectura del sistema, garantizando que cada componente funcione de manera óptima y cooperativa, lo que conduce a un diseño robusto y escalable.

los **Diagramas de Actividades y Componentes** mejoran de la comunicación que estandariza la manera en que los equipos de trabajo (desarrolladores, analistas, diseñadores) se comunican, asegurando que todos tengan una comprensión común de los sistemas y procesos, lo que minimiza malentendidos y errores, al tener una representación visual clara de los sistemas y procesos permite a los directivos y desarrolladores tomar decisiones más informadas respecto a cambios y mejoras en el sistema.

En la **Vida Cotidiana** el **Diagramas de Actividades** puede ser útil en la planificación y organización para organizar tareas personales o proyectos, ayuda a descomponer actividades en

pasos más pequeños y secuenciales, facilitando la gestión del tiempo y los recursos.

El manejo de UML, y en particular de los diagramas de actividades y componentes, fortalece tanto el Análisis y el Diseño de Sistemas, además de mejorar la eficiencia y la claridad en la gestión de proyectos en la vida laboral, asimismo, su utilidad se extiende a la planificación y solución de problemas cotidianos, aportando una metodología clara y estructurada.

es un **diagrama de componentes UML** titulado "Sorteo en Vacaciones en Familia", que representa la interacción entre los diferentes componentes del sistema que gestionan un sorteo vinculado a compras y transacciones. Aquí está la descripción general:

Componentes del sistema:

Cliente:

Este componente representa al cliente que realiza pedidos. El cliente interactúa con el sistema creando pedidos y realizando acciones como verificar el estado de los boletos emitidos para el sorteo.

Usuario:

Este componente agrupa las acciones generales de cualquier tipo de usuario (clientes, empleados, Administradores). Se conecta con el componente cliente y el componente empleado.

Pedido:

El pedido es creado por el cliente. Se muestra el flujo donde un cliente puede crear un pedido, que luego se conecta a las transacciones para verificar el pago.

Transacción:

Este componente maneja las transacciones de pago. Una vez que se crea un pedido, este pasa por el sistema de transacciones, que a su vez verifica el pago con el sistema bancario.

Sistema Bancario:

Este componente verifica el pago del pedido. Si el pago es exitoso, el sistema permite generar un ticket y proceder con el sorteo.

Sorteo:

Una vez que el pago ha sido verificado, el sistema emite boletos y los asigna al sorteo correspondiente. El componente sorteo está conectado a la emisión de tickets y a la verificación de boletos.

Ticket:

Este componente emite un ticket al cliente una vez que la transacción ha sido procesada. Está relacionado con el sorteo y la verificación de boletos.

Sistema de Verificación de Boletos:

Este componente verifica los boletos emitidos para el sorteo. Se conecta con el componente de sorteo y el sistema de pago de ganadores para verificar cuáles son los boletos ganadores.

Sistema Pago Ganadores:

Procesa el pago a los ganadores del sorteo, una vez que los boletos ganadores han sido verificados.

Administrador:

El Administrador gestiona el sistema, estableciendo los parámetros del sorteo, verificando que todo esté funcionando correctamente, y emitiendo notificaciones en caso de que haya algún problema.

Empleado:

El empleado también interactúa con el sistema, gestionando las actividades relacionadas con el sorteo, como la verificación de ganadores o la gestión de boletos emitidos.

Flujos de proceso:

El cliente **crea un pedido**, que luego es procesado a través del **sistema de transacciones**.

El **sistema bancario** se encarga de **verificar el pago**. Si es aprobado, se emite un **ticket** y el cliente es ingresado en el **sorteo**.

El sistema **procesa los ganadores** y **verifica los boletos** ganadores, asegurando que el sistema de sorteos emita correctamente los resultados.

Finalmente, el **sistema de pago de ganadores** distribuye los premios a los clientes con boletos ganadores.

Este diagrama describe cómo se gestiona el proceso completo del sorteo de “Vacaciones en Familia”, desde la creación del pedido, la verificación del pago, la emisión de boletos, la participación en el sorteo y el procesamiento de los ganadores, el flujo incluye tanto las interacciones entre el cliente y el sistema, como el rol de los Administradores y empleados en la gestión y verificación del sorteo.

Referencias:

- Ejemplo de diagrama de actividades con carriles* / Lucidchart. (s. f.). Lucidchart.
<https://www.lucidchart.com/pages/es/plantillas/ejemplo-de-diagrama-de-actividades-con-carriles>
- Oana, C. (2024, 12 marzo). *10+ plantillas de diagrama de carril y cómo crearlas*.
 Venngage Blog. <https://es.venngage.com/blog/diagrama-de-carril/>
- Planter. (2022, 30 julio). *Diagrama de componentes UML*. Plantilla Arbol Genealógico.
<https://plantillaarbolgenealogico.net/diagramas/componentes/>
- Recorridouml. (s. f.). *DIAGRAMAS DE COMPONENTES*.
<https://recorridouml.blogspot.com/2012/05/diagramas-de-componentes.html>
- Vpvera. (2022, 7 marzo). *Diagrama de actividad UML: una guía completa* - Cibermedio.
 Cibermedio. <https://www.cybermedian.com/es/uml-activity-diagram-a-comprehensive-guide/>