

Entornos de Desarrollo

Bloque 1

Tema 6: Análisis de una aplicación. Diagramas de flujo-Actividad

1.6.1. Introducción

Una vez establecidos los requisitos del sistema a desarrollar, se debe empezar a modelar la aplicación en sí. Una forma de hacerlo en un lenguaje estándar es mediante gráficos. Existen muchos diagramas gráficos para el análisis y diseño de aplicaciones y en este tema nos centraremos en uno de los primeros que se utilizaron.

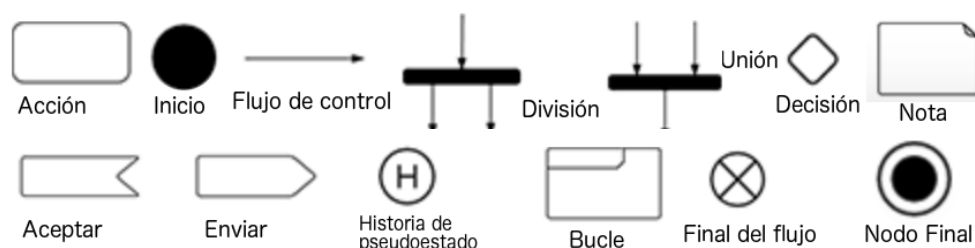
1.6.2 Diagrama de flujo

Un diagrama de flujo es una representación gráfica de un proceso. Cada paso del proceso se representa por un símbolo diferente que contiene una breve descripción de la etapa del proceso. Los símbolos gráficos del flujo del proceso están unidos entre sí con flechas que indican la dirección de flujo del proceso.

Este método gráfico de representación de procesos como pasos sucesivos de actividades ya se utilizaba en los años 30 del siglo pasado en la ingeniería industrial. En 1947 Herman Goldstine y John von Neumann lo desarrollaron para la aplicación en la planificación de programas de ordenador. Su última evolución se situaría en el diagrama de actividad del modelado UML.

1.6.3 Componentes de un diagrama de actividad

Los principales componentes de un diagrama de actividad son:

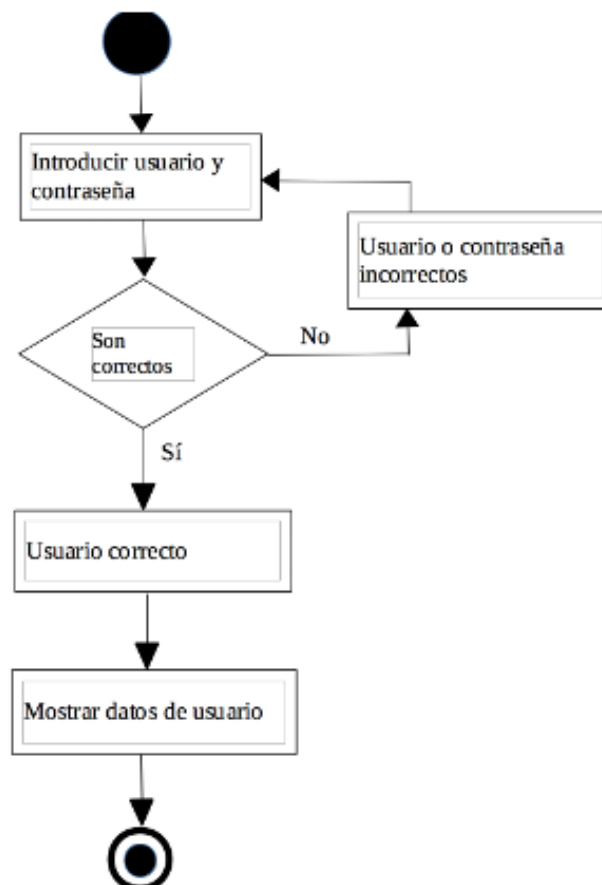


- **Acciones:** Un paso en una tarea dada. Se representa como un **rectángulo redondeado**.
- **Nodo de decisión:** Una condición que significa una bifurcación en el flujo del software. Se representa mediante un **rombo**.
- **Flujos de control:** Son los conectores que representan el flujo entre pasos del diagrama. Se representan con **flechas**.

- **Nodo inicial:** Simboliza el inicio del proceso. Se representa con un **círculo negro**.
- Cuando hay **actividades ocurriendo a la misma vez** se representa con una **barra**. Si se generan dos flujos sería de división y si dos tareas simultáneas vuelven a converger en una sería de unión.
- **Nota:** Permite a los creadores o colaboradores del diagrama comunicar mensajes adicionales que no encajan dentro del diagrama en sí.
- **Aceptar:** Demuestra la aceptación de un evento. Después de recibirse el evento, el flujo que viene de esta acción estará completado.
- **Enviar:** Complementario con el anterior. significa que una señal está siendo enviada a una actividad receptora.
- **Historia de pseudoestado superficial:** epresenta una transición que invoca el último estado activo.
- **Bucle:** permite modelar una secuencia repetitiva dentro del símbolo de opción de bucle.
- **Final de flujo:** muestra el punto final de un flujo de proceso.
- **Nodo final:** Simboliza el final de la actividad. Se representa con un **círculo negro dentro de un círculo**.

1.6.4 Ejemplos

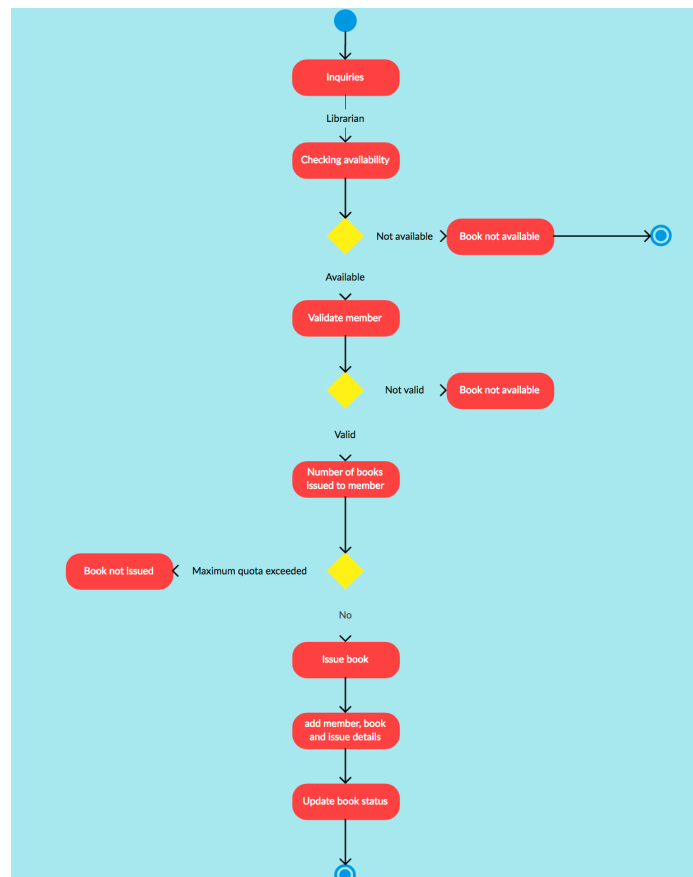
Validar usuario y contraseña



Como se puede ver en este ejemplo, reflejamos la lógica de la función de forma gráfica:

1. Se piden el nombre de usuario y contraseña
2. Se comprueba si son correctos o no
 - Si no son correctos volvemos al punto 1
 - Si sí son correctos
 - sacamos mensaje de usuario correcto
 - mostramos los datos de usuario.

Control de biblioteca

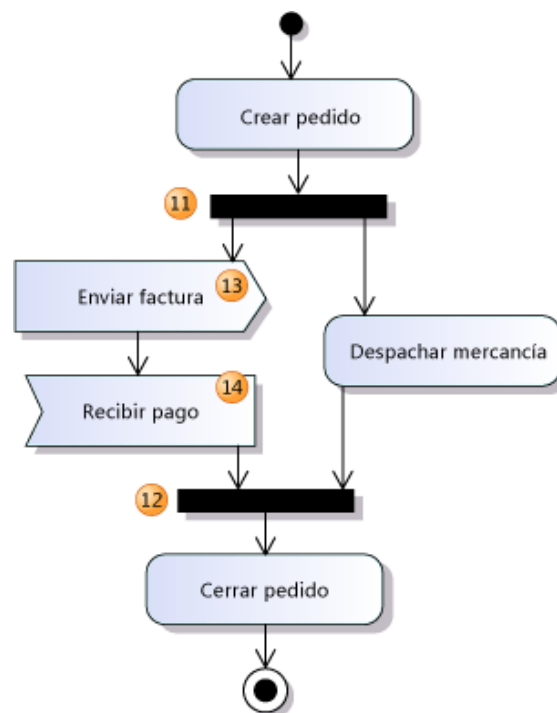


Este ejemplo representa la lógica de la búsqueda y petición de préstamo de un libro en una biblioteca.

1. Se busca el libro o libros en cuestión
2. Se comprueba la disponibilidad
3. Si no está disponible
 - Se muestra mensaje de no disponibilidad
 - Fin del proceso
4. Si está disponible
 - Se comprueba si el usuario es socio de la biblioteca
 - Si no es socio
 - Se muestra mensaje de no disponibilidad
 - Fin del proceso
 - Si es socio

- Se comprueba si se excede el número de libros prestados a este socio
 - Si no es correcto
 - Se muestra error de número de libros prestados
 - Fin del proceso
 - Si es correcto
 - Se presta el libro
 - Se crea el préstamo con los datos del miembro y el libro
 - Se actualiza el estado del libro

Pedidos



Ejercicios propuestos

1.6.4.1. Analiza el ejemplo de **Pedidos** y escribe la explicación del mismo siguiendo un modelo parecido al presentado en los ejemplos anteriores.

1.6.4.2. Realiza el diagrama de flujo de los siguiente enunciados sencillos usando el software *Dia* visto en el tema anterior:

1. Realiza un programa que pida un número y saque un mensaje indicando si es par o impar.
2. Realiza un programa que pida un número y lo siga pidiendo hasta que el número introducido sea impar.
3. Realiza un programa que vaya pidiendo un número hasta que se introduzca el número 0 y al final muestre el total de números positivos y negativos introducidos.
4. Realiza un programa que pida usuario y contraseña y si no es correcto los vuelva a pedir 3 veces. Al tercer intento si no es correcto se sacará el mensaje de incorrecto y se acabará el proceso.

1.6.4.3. Realiza el diagrama de actividad del ejercicio **1.5.3.4** del tema anterior.

1.6.5. Bibliografía

- [Tutorial de diagrama de actividades UML](#)
- [UML Diagram Types Guide: Learn About All Types of UML Diagrams with Examples](#)
- [Qué es un Diagrama de Flujo – Gestión de Procesos](#)
- [Unified Modeling Language \(UML\)](#)