



UNIVERSIDAD DE GUADALAJARA  
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E  
INGENIERÍAS



DEPARTAMENTO DE CIENCIAS COMPUTACIONALES  
SEMINARIO DE SOLUCIÓN DE PROBLEMAS DE INGENIERÍA DE  
SOFTWARE.

SECCIÓN: D01

PROFESORA: KARLA ÁVILA CÁRDENAS

CALENDARIO 2021B

**Práctica #9: Diseño lógico-físico**

Nombre: Ian Gerardo Horcasitas Pérez

Carrera: Ingeniería en Computación

Código: 216583618

## ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>3</b>
<b>DESARROLLO .....</b>	<b>4</b>
Diagrama de estado.....	4
Diagrama de actividades.....	5
Vista general.....	5
Proceso pantalla 1 .....	6
Proceso pantalla 2 .....	7
Proceso pantalla 3 .....	8
Diagrama de componentes .....	9
Diagrama de despliegue .....	9
<b>CONCLUSIONES .....</b>	<b>10</b>
<b>REFERENCIAS.....</b>	<b>10</b>

## INTRODUCCIÓN

Es importante realizar correctamente la planeación del software desde el principio hasta el momento previo a su construcción, pues los documentos generados servirán como base para el desarrollador. Los diagramas deben exponer con claridad y facilidad cada uno de los aspectos de lo que se desea construir, así como permitir la integración de nuevos requerimientos indicados por el cliente. También permite corregir errores que no fueron identificados en diagramas anteriores debido, pues se analiza el problema desde una perspectiva diferente.

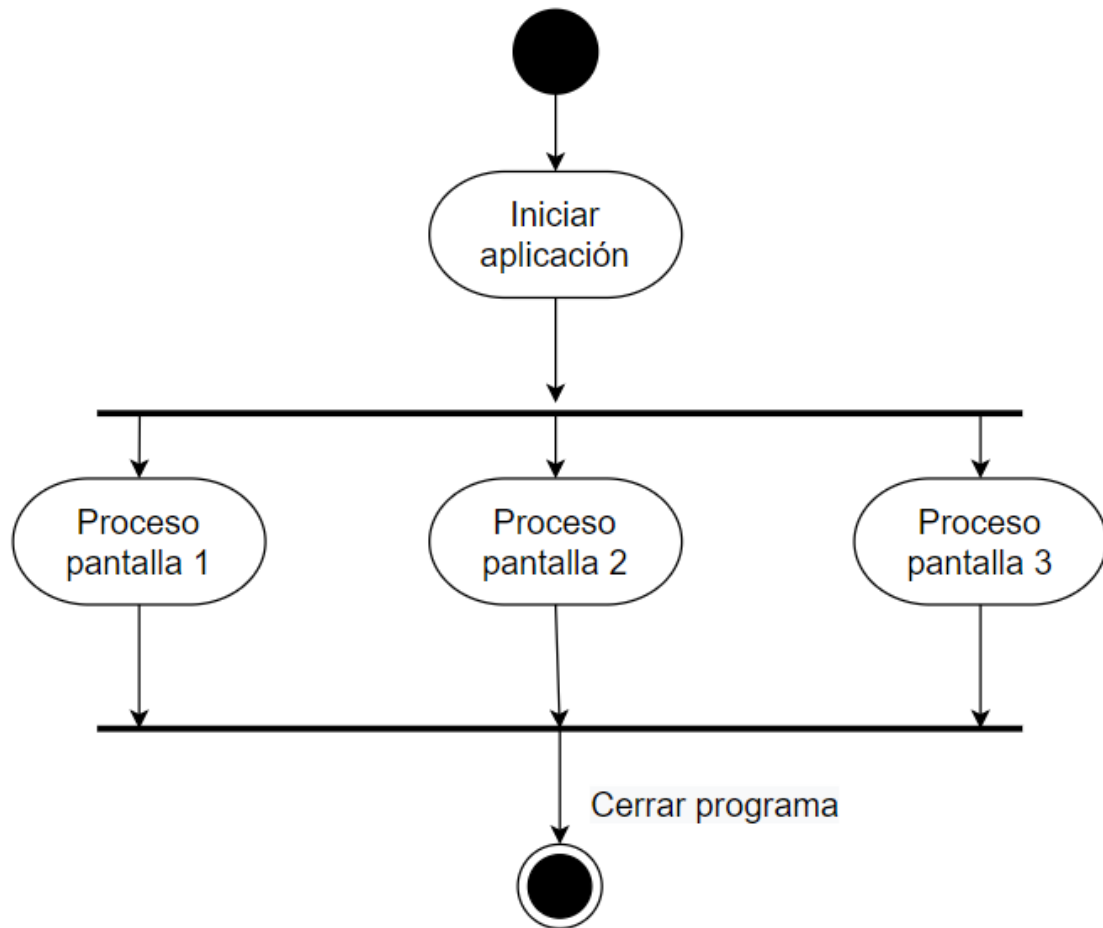
Previamente a la fase de creación, se realizarán 4 diagramas más puesto que se mostrará de una manera más explícita el funcionamiento del programa, los diferentes estados en los que se encontrará, los diferentes componentes que lo conformarán y la manera en que se dividirá al momento de convertirse en el producto final. Dichos diagramas son el diagrama de estado, el diagrama de actividades, el diagrama de componentes y el diagrama de despliegue respectivamente. En otras palabras, estos diagramas se encargarán de describir el funcionamiento lógico y físico del programa.

En primer lugar se encuentra el diagrama de actividad o actividades, el cual en palabras de Sommerville “Los diagramas de actividad intentan mostrar las actividades que incluyen un proceso de sistema, así como el flujo de control de una actividad a otra” (2011, p.123). Estos se encargan de representar la información ya vista en diagramas anteriores como el diagrama de secuencia y el de bloques en un solo diagrama. En segundo lugar, está el diagrama de actividades, similar al anterior, Summers (2011) menciona que su objetivo es mostrar la secuencia de las acciones que se realizan en un proceso. En tercer lugar, está el diagrama de componentes el cual busca mostrar los componentes que formarán al programa y sus dependencias entre ellos. Finalmente se tiene al diagrama de despliegue, el cual es uno de los más importantes pues se encarga de representar la manera en que el sistema será repartido, indicando la localización de cada componente en cada parte dividida del programa, así como las diferentes conexiones que existirán entre los fragmentos para comunicarse y hacer funcionar al producto final en su totalidad.

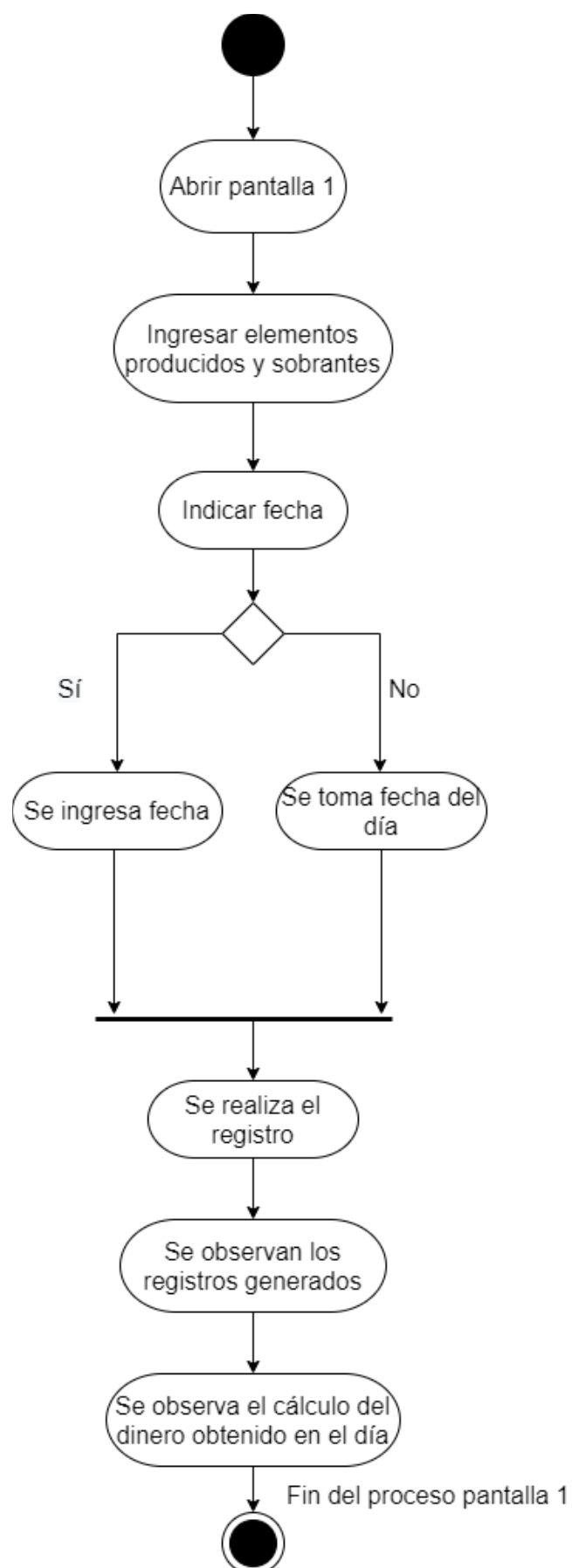


## Diagrama de actividades

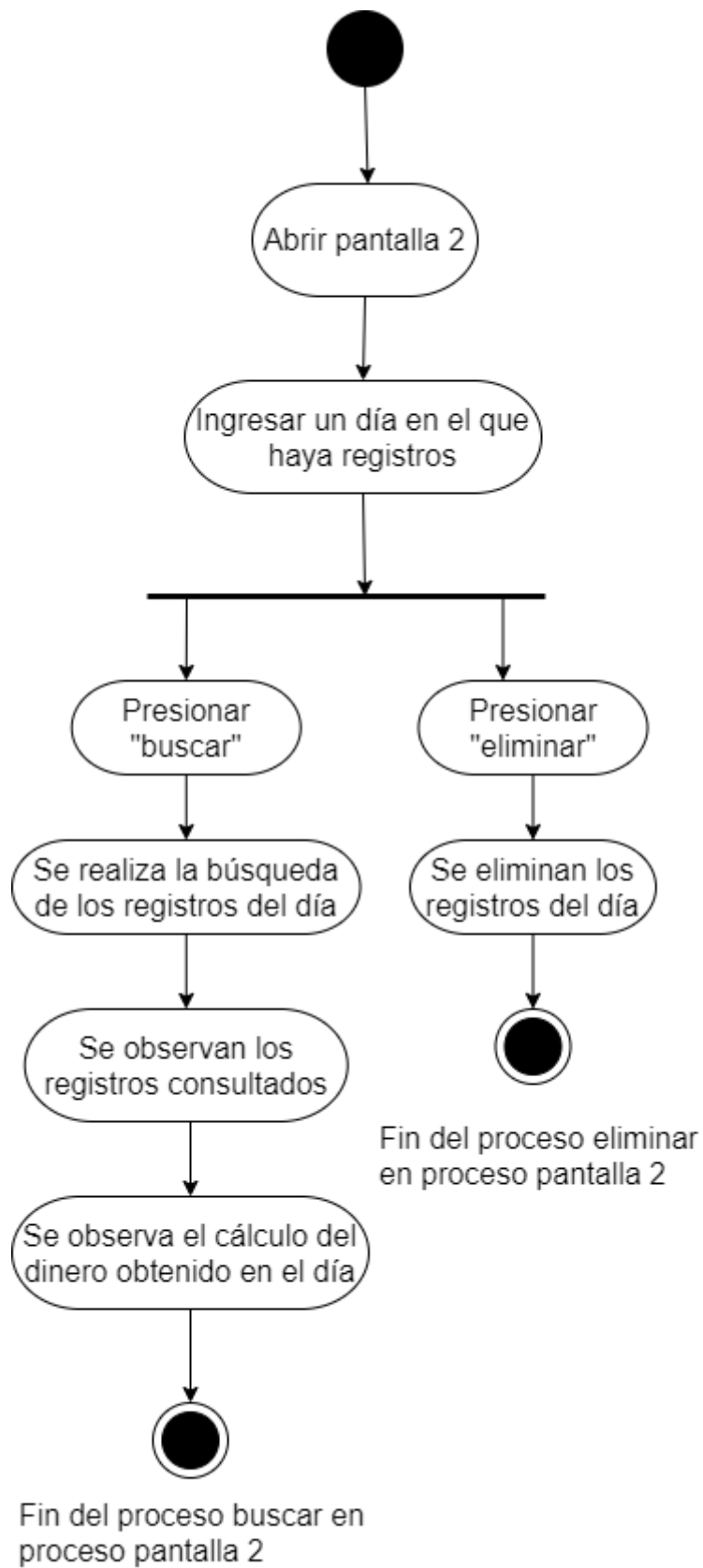
Vista general



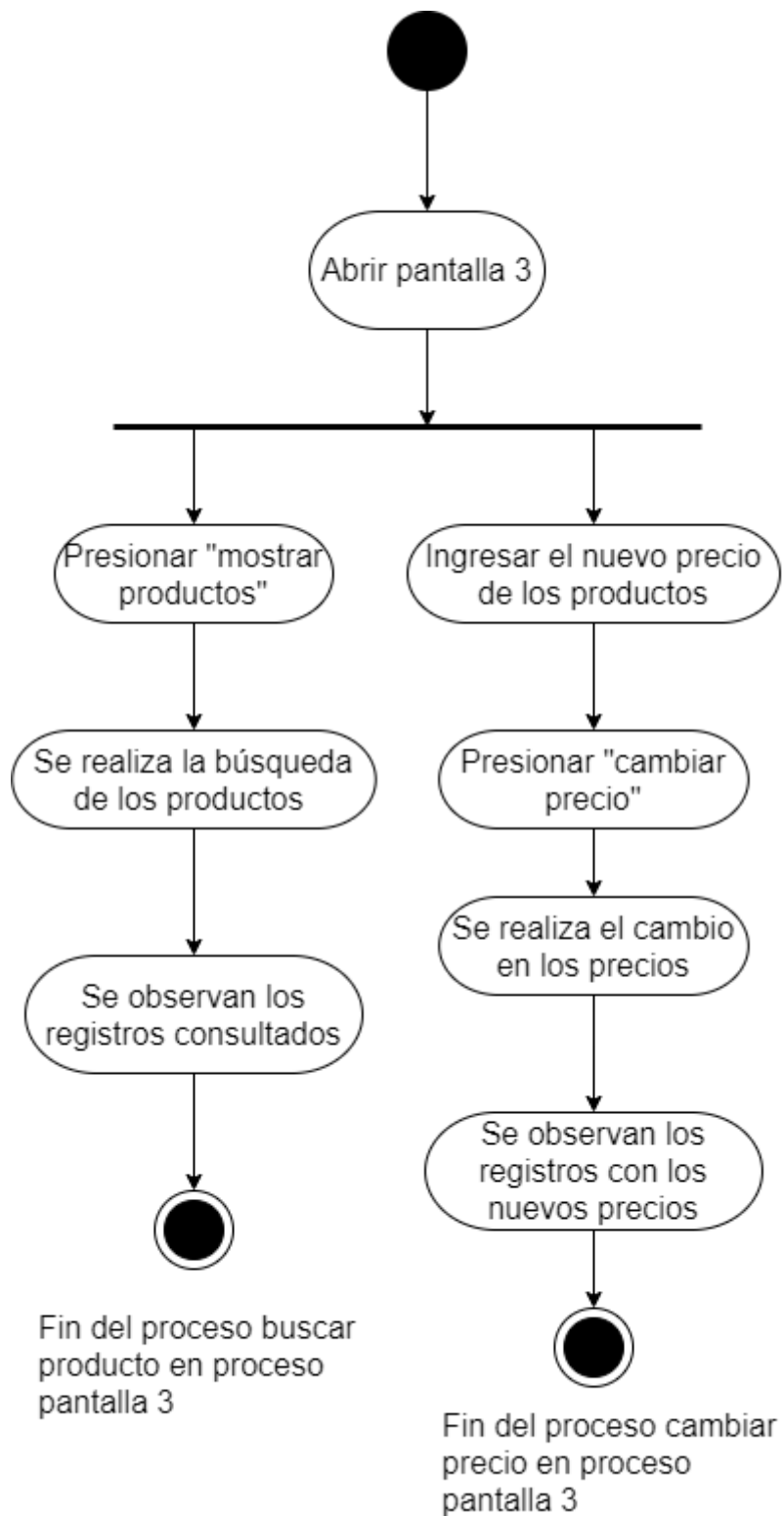
## Proceso pantalla 1



## Proceso pantalla 2



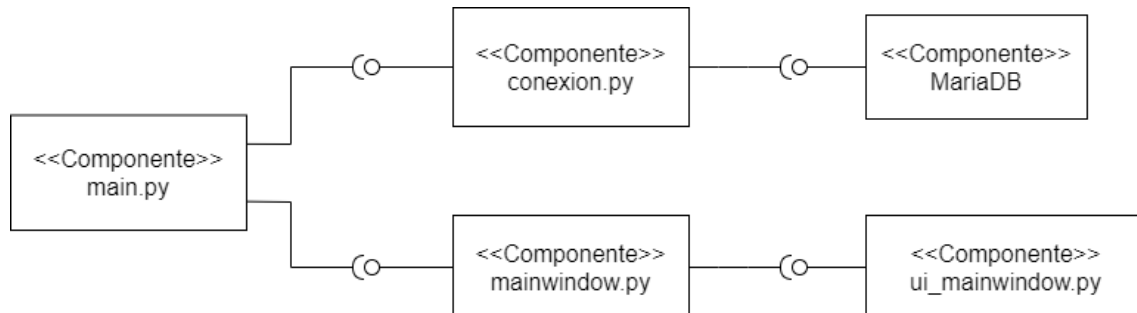
### Proceso pantalla 3



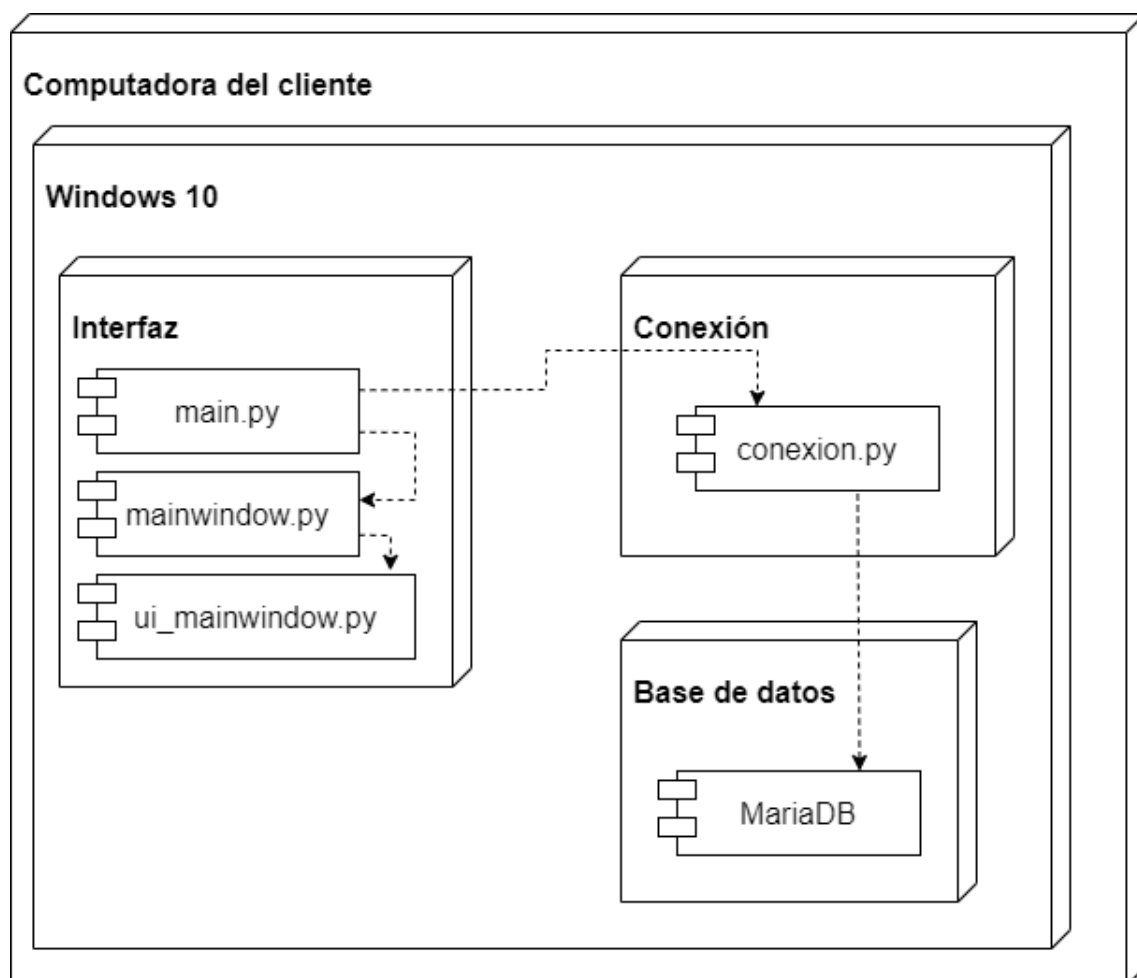


### Diagrama de componentes

Debido a que no se utilizarán clases en el programa se incluirá en el diagrama de componentes todos los archivos de código que se tienen planeados utilizar en el programa, así como la base de datos.



### Diagrama de despliegue



## **CONCLUSIONES**

Encontré bastante útil esta práctica, pues con ella pude descubrir algunos problemas menores que podrían aparecer en mi programa, sin embargo, no lo sabre con total seguridad hasta realizar la implementación. Aún así, el haber realizado toda la planeación desde el principio, creado los diagramas e ideando la construcción del programa desde cero evité una gran cantidad de posibles errores. Es debido a esto que podré realizar el sistema entero por mi cuenta en un plazo de dos semanas como máximo, de otra forma, mi progreso se vería disminuido en gran medida por correcciones o cambios que terminarían por afectar todo el programa en general, teniendo que corregir una innumerable cantidad de situaciones que se conectan a un solo error.

Aún aplicando los métodos de ingeniería de software nunca se tiene total certeza que el programa se podrá desarrollar de la manera en que fue planeado desde un principio, pues siempre estará abierto a cambios y modificaciones necesarias para cubrir todas las necesidades del cliente y arreglas problemas que no fueron contemplados.

## **REFERENCIAS**

Sommerville, I. (2011). Ingeniería del software (9° ed). PEARSON EDUCACIÓN.