Análisis y modelado de un sistema para la gestión de formularios

López, Carlos – 2016202

López González, Brayan Arnulfo – 20162021313

1 de marzo de 2019

Universidad Distrital Francisco José de Caldas

Ingeniería de Sistemas

Fundamentos de Ingeniería de Software

Bogotá, D. C.

Tabla de contenido

[Introducción 3](#_Toc2123051)

[Objetivos 4](#_Toc2123052)

[General 4](#_Toc2123053)

[Especifico 4](#_Toc2123054)

[Análisis y modelado 5](#_Toc2123055)

[Planteamiento de la situación 5](#_Toc2123056)

[Organización del proyecto 6](#_Toc2123057)

[Diagrama de casos de uso 6](#_Toc2123058)

[Diagrama de clases 8](#_Toc2123059)

[Conclusiones 9](#_Toc2123060)

# Introducción

Existen varias estructuras de datos que ayudan y facilitan el trabajo.

# Objetivos

## General

Realizar un sistema para facilitar la gestión de formularios en una empresa.

## Especifico

Aplicar los conceptos y habilidades adquiridas durante el semestre.

# Análisis y modelado

## Planteamiento de la situación

Una empresa desea recoger información acerca de sus empleados con el fin de hallar perfiles que les permitan enfocar el desarrollo de las labores de manera más eficiente. Para ello, los gerentes ordenan la elaboración de formularios que son entregados en cada una de las sucursales ubicadas a lo largo y ancho del país (puede haber más de una sucursal en la misma ciudad). La información que se debe consignar allí es la siguiente:

* Nombre
* Apellido
* Tipo de identificación: puede ser uno de los siguientes valores: Cédula de ciudadanía, cédula de extranjería o tarjeta de identidad.
* Número de identificación
* Sexo: F o M
* Teléfono celular
* Teléfono fijo
* e-mail
* Fecha de Nacimiento
* Ciudad de Nacimiento
* País de Nacimiento
* Ciudad de Residencia
* Dirección
* Barrio
* Actividad laboral: Artes, ciencias sociales, ingenierías, áreas de la salud, otros
* Tiene hijos?: como respuesta se acepta S ó N
* Número de hijos
* Listado con el nombre y fecha de nacimiento de cada uno de los hijos: 0-5, 6-10, 11-15, mayor de 16
* Nombre de la sucursal en la que trabaja

Datos de la sucursal.

* Nombre de la sucursal
* Dirección de la sucursal
* Barrio
* Nombre del gerente de la sucursal.

Para evitar que la solicitud de información se haga demasiado grande, el programa debe estar en capacidad de calcular la edad de cada una de las personas encuestadas.

## Requerimientos funcionales

Con el fin de optimizar las búsquedas deberán utilizarse listas y la multilista definida. Las cuales se construyen al iniciar el programa y a partir de la información almacenada en los archivos.

Debe garantizar la eficiencia en las búsquedas aprovechando las estructuras de datos que considere pertinentes.

El programa presenta un menú y submenús que facilitan las consultas.

Adicionalmente, las listas deben crearse de manera genérica, de tal forma que puedan agregarse fácilmente otras consultas adicionales por parte de la empresa.

El programa debe estar en capacidad de calcular la edad de cada una de las personas registradas.

Para la entrega y sustentación final deben incluir los archivos planos indispensables con no menos de 30 registros.

Debe aplicar programación orientada a objetos y el principio de alta cohesión y bajo acoplamiento.

Debe aplicar los conceptos de optimización estudiados en el curso. No limitarse a garantizar el funcionamiento del programa.

Al finalizar el programa se actualizan los archivos incluyendo los cambios realizados durante la ejecución. Se agregaron o retiraron feligreses.

## Organización del proyecto

El proyecto usa la arquitectura de modelo-vista-controlador. En el modelo están definidas las estructuras de datos que servirán para la gestión de formulario. La vista es presentada por consola, esta permite al usuario poder acceder a las diferentes funcionalidades del programa. El controlador verifica los datos ingresados por el usuario y conecta la vista con el modelo.

## Diagrama de casos de uso

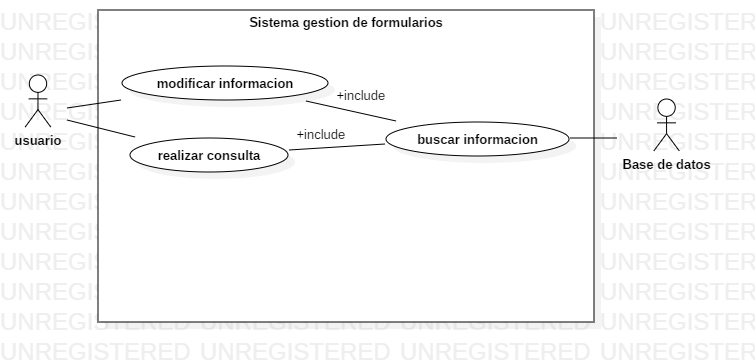


Fig. 1. Diagrama de casos de uso. Fuente: elaboración propia en StarUML.

En la figura 1 se observa el diagrama de casos de uso. A continuación se encuentran los casos de uso detallados y los casos de uso generales.

|  |  |
| --- | --- |
| Caso de uso | RegistrarPresentaciones |
| Identificador | 001 |
| Actor | Administrador |
| Descripción | El administrador puede registrar una nueva presentación teatral en el cronograma de presentaciones, para que aparezca entre las disponibles para el público. |
| Precondición | Debe haber una presentación sin registrar. |
| Flujo de eventos | |
| Actor | Sistema |
| 1. El administrador solicita el registro de una presentación.    3. El administrador selecciona una sala y un horario a la presentación dependiendo de la disponibilidad.    6. El administrador confirma el registro de la presentación. | 2. El sistema muestra el cronograma, donde aparece la organización de las presentaciones y las salas y horarios disponibles.    4. El sistema comprueba que la selección del administrador sea correcta.  5. El sistema solicita confirmación del registro de la presentación.  7. El sistema realiza el registro de la presentación en el cronograma de presentaciones. |
| Postcondición | El administrador ha registrado una nueva presentación que aparecerá en el cronograma estará disponible para el público. |

## Diagrama de clases

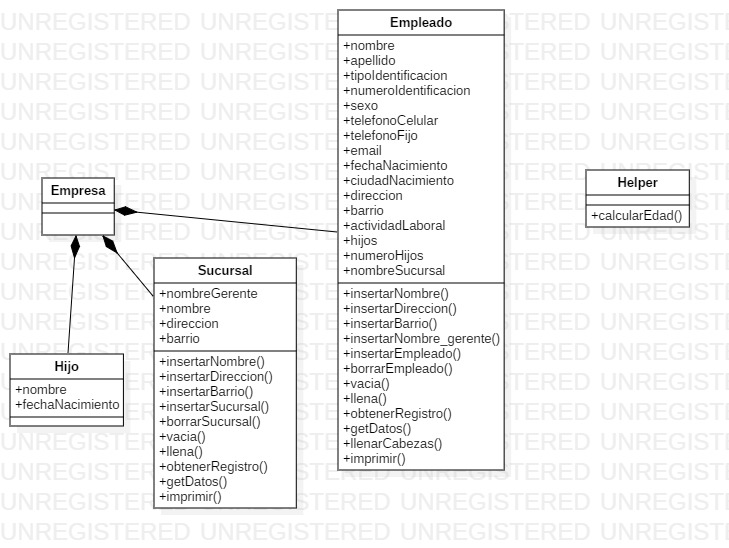


Fig. 2. Diagrama de Clases. Fuente: elaboración propia en StarUML.

Como se puede ver en la figura 2, el proyecto consta de cinco clases. La clase empresa que contiene a las clases empleado, sucursal e hijo.

# Conclusiones

Usar estructuras de datos como multilistas permite estructurar un proyecto de forma adecuada para realizar consultas y hacer las modificaciones correspondientes de forma rápida.