



**MÉTODO DE DISEÑO DE LA INGENIERÍA  
COMPUTACIÓN Y ESTRUCTURAS DISCRETAS I  
TAREA INTEGRADORA II**

**By:**

**Juan David Bahamon Rodriguez**

**A00375826**

**Carlos Javier Bolaños Riascos**

**A00377995**

**Samuel Hernandez Espitia**

**A00375392**

**David Esteban Peñaranda Scarpetta**

**A00375827**

## **FASE 1: IDENTIFICACIÓN DEL PROBLEMA**

El equipo VIP de simulación de la Facultad de Ingeniería de la Universidad Icesi, ha contactado a nuestro equipo de desarrolladores en formación para ser parte de su subproyecto de investigación. Nuestra tarea asignada es el diseño e implementación de un prototipo de software que permita la gestión eficiente de operaciones CRUD (Create, Read, Update and Delete ) sobre una base de personas de nuestro continente. Los datos a generar son: código (que es autogenerado por nuestro proyecto de software), nombre, apellido, sexo, nacionalidad y fotografía.

La cantidad de datos a gestionar en nuestro proyecto es 1 millón, por lo que debemos construir un modelo eficiente, relacionando las buenas prácticas de la ingeniería de software y las estructuras de datos aprendidas en el curso de “**Computación y Estructuras Discretas I**”. Además, se nos ha pedido buscar una solución para permitir la interacción intuitiva y cómoda entre el software y el usuario (Implementación de una interfaz gráfica).

Las condiciones entregadas por el equipo VIP de Simulación son:

1. El registro de un millón de personas debe ser generado con dos bases de datos que han sido puestas a nuestra disposición, que contienen nombres y apellidos, los cuales deben ser combinados todos los nombres con todos los apellidos.
2. Se debe de generar de forma aleatoria la fecha de nacimiento de cada una de las personas que nacen en el continente americano.
3. Teniendo en cuenta una las estaturas promedio de los hombres y de las mujeres, se debe de generar aleatoriamente la estatura de cada una de las personas según su género y que esta tenga sentido.
4. Cada persona al nacer tiene una nacionalidad, por lo que debe ser generada teniendo en cuenta los porcentajes relativos de la población de cada país con respecto al continente americano, en caso de que las nacionalidades generadas sean diferentes a la población total, se debe de realizar una distribución del sobrante o faltante de tal forma que quede la misma población total.
5. La fotografía de la persona debe ser generada aleatoriamente en un sitio web, teniendo en cuenta que no importa el género, edad, estatura y otros.

En cuanto a la funcionalidad que ofrece el programa al usuario son las siguientes:

1. El programa debe contener una barra de progreso, el cual en caso de que la operación se demore más de 1 segundo debe de indicar el tiempo que se demoró en finalizar.
2. El usuario tiene la posibilidad de elegir cuántos registros desea generar, para ello se dispone de un campo de texto el cual permite digitar el número de registros, teniendo en cuenta que hay un valor máximo.
3. El programa debe contar con una opción para guardar los datos generados en la base de datos, permitiendo realizar futuramente una consulta.
4. La interfaz es amigable y brinda las siguientes funcionalidades al usuario:
  - Dispone de campos de texto los cuales permiten al usuario agregar una nueva persona a la base de datos.
  - Permite que el usuario realice la búsqueda de una persona, seleccionando un solo criterio, bien sea nombre, apellido, nombre completo o código.
  - Dispone de campos de texto que permiten actualizar datos de una persona existente.
  - Permite seleccionar una persona y eliminarla de la base de datos.
  - Para la funcionalidad de búsqueda con los criterios de nombre, apellido y nombre completo, a medida que vaya el usuario ingresando las letras, cuando coincidan menos de 20 elementos, debe de aparecer una lista emergente con los nombres de la base de datos que coinciden.

#### **Análisis de requerimientos:**

<b>Cliente</b>	Equipo VIP de simulación de la Universidad Icesi.
<b>Usuario</b>	Integrantes del Equipo VIP de simulación de la Universidad Icesi.
<b>Contexto del problema</b>	<p>El equipo VIP de simulación de nuestra universidad, nos ha contactado para asignarnos la tarea de hacer un prototipo de software que haga gestión de un dataset con <b>Operaciones CRUD</b>. Nuestro proyecto debe tener la capacidad de gestionar 1 millón de datos.</p> <p>Para cumplir con nuestra tarea, debemos conocer estadísticas que nos ayuden a</p>

	<p>generar datos en rangos coherentes. Al ser una simulación de datos planteada para el continente americano, debemos usar nacionalidades que pertenezcan a América. Por otra parte, necesitamos datos base (nombre y apellidos) para iniciar el proceso de creación y simulación.</p>
<b>Requerimientos funcionales</b>	<p>GEDT1: Importar de los datasets suministrados.</p> <p>GEDT2: Generar datos.</p> <p>UIT1: Permitir generación de datos.</p> <p>GEDT3: Guardar datos generados.</p> <p>ALDT1: Almacenar los datos (Persistencia).</p> <p>UITI2: Permitir creación de datos.</p> <p>UITI3: Permitir búsqueda de datos.</p> <p>UITI4: Permitir actualización/edición de datos.</p> <p>UITI5: Permitir eliminación de datos.</p> <p>GEDT4: Crear datos.</p> <p>GEDT5: Consultar datos.</p> <p>GEDT6: Actualizar/Editar datos</p> <p>GEDT7: Eliminar datos.</p>
<b>Requerimientos no funcionales</b>	<p>El prototipo de software debe estar construido e implementado en el lenguaje de programación Java. Además de contener una interfaz gráfica con elementos de Javafx y pruebas unitarias.</p> <p>El manejo de los datos, deben estar ordenados y balanceados siguiendo la teoría de los Arboles ABB y AVL.</p>

<b>Identificador</b>	GEDT1: Importar de los datasets suministrados		
<b>Resumen</b>	Dados los datasets de nombres, apellidos, nacionalidad, distribución de edad y estatura el sistema debe leerlos para que más adelante poder generar y combinar los datos a petición del usuario.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	nombre chicos	File (csv)	
	nombre chicas	File (csv)	
	apellidos	File (csv)	
	distribución edad	File (csv)	
	distribución estatura	File (csv)	
	nacionalidad	File (csv)	
<b>Actividades generales necesarias para obtener resultados</b>	Se leen los datasets.		
<b>Resultado o postcondición</b>	Lectura de los datasets.		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>

<b>Identificador</b>	GEDT2: Generar datos.		
<b>Resumen</b>	El programa debe de tomar los datasets de nombres y apellidos, edades, estaturas y nacionalidad, para poder hacer uso de ellos y generar cada una de las personas en la base de datos.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	nombre chicos	File (csv)	
	nombre chicas	File (csv)	
	apellidos	File (csv)	
	edad	int	
	estatura	int	
	género	String	
	nacionalidad	String	
<b>Actividades generales necesarias para obtener resultados</b>	<ol style="list-style-type: none"> <li>1. El programa debe de tomar los datasets de los nombres y apellidos, los cuales deben ser combinados.</li> <li>2. Tomar los datasets de la distribución de las edades, estaturas y nacionalidad para generar los datos necesarios para crear una persona.</li> </ol>		
<b>Resultado o postcondición</b>	Los datos de cada una de las personas ha sido generado exitosamente.		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	nombre	String	
	apellidos	String	
	edad	int	
	estatura	int	
	género	String	
	nacionalidad	String	

<b>Identificador</b>	UIT1: Permitir generación de datos.		
<b>Resumen</b>	La interfaz del prototipo debe tener un módulo que permita iniciar el proceso de generación de datos. Para iniciar la generación, el usuario ingresa la cantidad de registros a generar. Además, se debe mostrar una barra de proceso que indique cómo va la solicitud realizada por el usuario.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	cantidad de registros a generar	int	
<b>Actividades generales necesarias para obtener resultados</b>	<ol style="list-style-type: none"> <li>1. Leer cantidad de registros</li> <li>2. Iniciar proceso de generación</li> <li>3. Mostrar barra de proceso</li> </ol>		
<b>Resultado o postcondición</b>	Datos/Registros generados		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	Registro	Person	Es de tipo Person, ya que al generar los datos, en el backend lo que se hace es crear objetos de tipo Person con estos datos.

<b>Identificador</b>	GEDT3: Guardar datos generados.		
<b>Resumen</b>	El programa debe de permitir tomar cada uno de los datos generados y guardarlos en una base de datos.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	Registro	Person	Se guarda el objeto de tipo Person. Ahí están los datos generados por el sistema que deseamos guardar.
<b>Actividades generales necesarias para obtener resultados</b>	Cuando se realiza la generación de los datos de la persona, el sistema debe de tomar esos datos y guardarlos en una base de datos.		
<b>Resultado o postcondición</b>	Los datos han sido guardados exitosamente.		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>



<b>Identificador</b>	ALDT1: Almacenar los datos (Persistencia).		
<b>Resumen</b>	El sistema debe de tomar todos los datos que han sido generados y cuando el usuario pulse el botón de guardar o se cierre la pestaña de la interfaz, se deben de guardar cada uno de los datos, de tal forma que en el futuro cuando se abra el programa, estos datos sigan a disposición del usuario.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	nombre	String	El campo ha sido generado
	apellidos	String	El campo ha sido ingresado
	edad	int	El campo ha sido generado
	estatura	int	El campo ha sido generado
	género	String	El campo ha sido generado
	nacionalidad	String	El campo ha sido generado
<b>Actividades generales necesarias para obtener resultados</b>	<ol style="list-style-type: none"> <li>1. Al pulsar el botón de guardado, el sistema guarda todos los datos (persistencia).</li> <li>2. Cuando el usuario vuelva a abrir el programa, todos los datos que han sido previamente guardados, permanecerán en el programa.</li> </ol>		
<b>Resultado o postcondición</b>	Los datos han sido guardados exitosamente y permanecen en el programa para un futuro uso.		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>

<b>Identificador</b>	UITI2: Permitir creación de datos.		
<b>Resumen</b>	El sistema debe permitir al usuario ingresar al módulo de crear. Ahí, el usuario ingresará los datos necesarios en los campos solicitados para la creación de una persona en la base de datos.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	nombre	String	
	apellidos	String	
	edad	int	
	estatura	int	Se deben ingresar en centímetros
	nacionalidad	String	
	género	String	
<b>Actividades generales necesarias para obtener resultados</b>	<ol style="list-style-type: none"> <li>1. Leer los datos ingresados en los campos.</li> <li>2. Pasar los datos como parámetros a los métodos para la creación del objeto persona. (Ver en GEDT4)</li> </ol>		
<b>Resultado o postcondición</b>	Registro creado.		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	Notificación "Registro completado"	Alerta	

<b>Identificador</b>	UITI3: Permitir búsqueda de datos.		
<b>Resumen</b>	La interfaz contiene una opción para seleccionar el criterio deseado, y posteriormente un campo de texto el cual permite al usuario realizar la búsqueda. A medida que el usuario va ingresando el texto, cuando haya menos de 20 coincidencias, se despliega una lista con recomendaciones de los usuarios.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	criterio	acción event	
	campoTexto	String	El criterio ha sido seleccionado
<b>Actividades generales necesarias para obtener resultados</b>	<ol style="list-style-type: none"> <li>1. La interfaz brinda un espacio de selección del criterio.</li> <li>2. La interfaz brinda un campo de texto.</li> <li>3. La interfaz brinda un campo de visualización de la lista de recomendaciones.</li> <li>4. La interfaz brinda un espacio para visualizar la persona que han sido buscada</li> </ol>		
<b>Resultado o postcondición</b>	La búsqueda de la persona ha sido exitoso o fracasado		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	person	Person	

<b>Identificador</b>	UITI4: Permitir actualización/edición de datos.		
<b>Resumen</b>	El sistema debe permitir al usuario ingresar al módulo de edición para actualizar los datos de una persona. Para eso, el usuario debe seleccionar el registro a editar, para poder sobrescribir los datos que este vea necesarios.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	nombre	String	
	apellidos	String	
	edad	int	
	estatura	int	se debe ingresar en centímetros
	género	String	
	nacionalidad	String	
<b>Actividades generales necesarias para obtener resultados</b>	<ol style="list-style-type: none"> <li>1. Leer los cambios realizados</li> <li>2. Pasar los datos cómo parámetros a los métodos para la actualización/edición del objeto persona. (Ver en GEDT6)</li> </ol>		
<b>Resultado o postcondición</b>	Registro actualizado.		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	Notificación “Actualización realizada con éxito”	Alert	

<b>Identificador</b>	UITI5: Permitir eliminación de datos.		
<b>Resumen</b>	El sistema debe permitir al usuario ingresar el módulo de eliminar el registro de una persona. Para eso, el usuario selecciona un usuario y da inicio al proceso al seleccionar el botón “Eliminar registro”.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	registro	Person	
<b>Actividades generales necesarias para obtener resultados</b>	1. Seleccionar el registro a eliminar. 2. Pasar el registro como parámetro a los métodos para la eliminación del objeto persona. (Ver en GEDT7)		
<b>Resultado o postcondición</b>	Registro eliminado.		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	Notificación “Registro eliminado con éxito”	Alert	

<b>Identificador</b>	GEDT4: Crear datos.		
<b>Resumen</b>	Iniciado el proceso de creación de registro desde el módulo de creación (Interfaz gráfica : UITI2). El sistema debe crear un registro. El id de cada persona, es autogenerado por el sistema, este es importante para que se pueda finalizar la creación.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	nombre	String	
	apellidos	String	
	edad	int	
	estatura	int	
	nacionalidad	String	
	género	String	
<b>Actividades generales necesarias para obtener resultados</b>	1. Se crea en nuevo registro 2. Se guarda en la base de datos.		
<b>Resultado o postcondición</b>	Registro creado.		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	insert	boolean	True = Se agregó False = No se agregó.

<b>Identificador</b>	GEDT5: Consultar datos.		
<b>Resumen</b>	Iniciado el proceso de búsqueda de un registro (guardado con anterioridad) desde el módulo de consulta (Interfaz gráfica : UITI3), el sistema debe buscar en su base de datos los registros similares que encajen con el criterio de búsqueda seleccionado por el usuario.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	criterio de búsqueda	String	
<b>Actividades generales necesarias para obtener resultados</b>	1. Buscar registros similares 2. Conexión con interfaz gráfica 3. Mostrar aproximaciones de la búsqueda.		
<b>Resultado o postcondición</b>	Registro seleccionado. (Se busca entre las posibles y se selecciona).		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	Registro	Person	

<b>Identificador</b>	GEDT6: Actualizar/Editar datos		
<b>Resumen</b>	Iniciado el proceso de actualización de un registro (guardado con anterioridad) en el módulo de edición (Interfaz gráfica : UITI4). El sistema debe guardar los cambios realizados al registro seleccionado por el usuario.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	nombre	String	
	apellidos	String	
	edad	int	
	estatura	int	se debe ingresar en centímetros
	género	String	
	nacionalidad	String	
<b>Actividades generales necesarias para obtener resultados</b>	1. Guardar los cambios.		
<b>Resultado o postcondición</b>	Registro actualizado		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	actualización	boolean	False = No se actualizó. True = Se actualizó.



<b>Identificador</b>	GEDT7: Eliminar datos.		
<b>Resumen</b>	Iniciado el proceso de eliminación de registro (previamente guardado) en el módulo eliminar. El sistema debe eliminar el registro seleccionado por el usuario.		
<b>Entradas</b>	<b>Nombre de la entrada</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	registro	Person	
<b>Actividades generales necesarias para obtener resultados</b>	1. Eliminar registro.		
<b>Resultado o postcondición</b>	Registro eliminado.		
<b>Salidas</b>	<b>Nombre de la salida</b>	<b>Tipo de dato</b>	<b>Condición de repetición o selección</b>
	delete	boolean	False = Si no se eliminó. True = Si se eliminó

**FASE 2: RECOPIACIÓN DE INFORMACIÓN NECESARIA**

**FASE 3: BÚSQUEDA DE SOLUCIONES CREATIVAS**

**FASE 4: TRANSICIÓN DE LA FORMULACIÓN DE IDEAS A LOS DISEÑOS  
PRELIMINARES**

**FASE 5: EVALUACIÓN Y SELECCIÓN DE LA MEJOR SOLUCIÓN**