|  |  |  |
| --- | --- | --- |
| **Objetivos**   1. Generar la habilidad para la algorítmica para localizar elementos en una estructura enlazada sencilla | **Número de participantes**  1 – Antony Salcedo | **Materiales**   * Internet * Eclipse |
| **Duración de la actividad**  **01:30** |
| **INSTRUCCIONES**   1. Ingresa a la plataforma institucional descarga el ejercicio Central de Pacientes, crea el proyecto en eclipse, ejecútalo y verifica que todas las opciones trabajan correctamente. **No olvides activar los assert en Eclipse.** 2. **[0.1]** Estudia el modelo conceptual del mundo de la aplicación y responde las siguientes preguntas:  |  |  | | --- | --- | | Pregunta | Respuesta | | ¿Cómo se llama el atributo de la clase CentralPacientes que hace referencia al primer paciente en la lista? | **Primero.** | | ¿De qué tipo es el anterior atributo? | Paciente. | | ¿Cómo se llama el atributo de la clase Paciente que hace referencia al paciente que sigue en la lista? | Siguiente. | | ¿De qué tipo es el anterior atributo? | Paciente. |  1. **[0.1]** Estudia el método darPacientes() de la clase CentralPacientes y responde las siguientes preguntas:  |  |  | | --- | --- | | Pregunta | Respuesta | | ¿Cuál es el objetivo del método? | Devuelve la lista del total de pacientes registrados actualmente. | | ¿Cómo se recorren los elementos de una lista enlazada? | Podemos tomar el mismo ejemplo del método mencionado en la pregunta anterior, donde se necesita recorrer una lista de pacientes, en la cual se hace unos de while como condicional para realizar el recorrido. | | ¿Para qué se utiliza la variable actual? | Según el método trabajado anteriormente en las preguntas, la variable actual se utiliza en el método **darPacientes()** para recorrer una lista enlazada de pacientes y agregar cada uno de ellos a una lista de pacientes (**ArrayList pacientes**).  Inicialmente, la variable actual se establece en el primer nodo de la lista enlazada (**Paciente primero**). Luego, se inicia un bucle **while** que recorre la lista enlazada mientras el paciente actual no sea **null**.  Dentro del bucle **while**, se agrega el objeto actual a la lista de pacientes (**pacientes.add(actual)**), y luego se actualiza la variable actual con el siguiente nodo de la lista enlazada (**actual = actual.darSiguiente()**).  De esta manera, se va recorriendo la lista enlazada y agregando cada paciente a la lista pacientes. Finalmente, se devuelve la **lista pacientes** con todos los pacientes de la lista enlazada. | | ¿De qué tipo es la variable actual? | La variable actual es de tipo **Paciente**, ya que se utiliza para almacenar el paciente actual de una lista enlazada de pacientes. En otras palabras, la variable actual almacena el objeto Paciente que se encuentra en el paciente actual de la lista enlazada. | | ¿Cuál es la condición de salida del ciclo? | La condición de salida del ciclo **while** es que la variable actual sea **null**. | | ¿Cómo avanza el ciclo? | El ciclo avanza utilizando el método **darSiguiente()** de la clase Paciente, que devuelve el siguiente paciente de la lista enlazada. | | ¿Por qué se utiliza un ciclo while para recorrer la lista enlazada y no un for? | Esto se debe a que se esta trabajando con referencias, mas no con índices. Por lo que el recorrido se hace a base de referencias, dado que en se fuese con índices se pueden utilizar ambas condicionales **while** y **for**. |  1. **[0.2]** Se necesita calcular y devolver la cantidad de pacientes que existen en la central de pacientes. Si no existen pacientes debe lanzarse el error correspondiente. Complete el requerimiento funcional.  |  |  |  |  | | --- | --- | --- | --- | | ID: | RF1 | Nombre: | Calcular Y Devolver La Cantidad De Pacientes Que Existen En La Central De Pacientes | | Resumen: | | | Se necesita calcular y devolver la cantidad de pacientes que existen en la central de pacientes. Si no existen pacientes debe lanzarse el error correspondiente. | | Entradas: | | | N/A | | Resultado: | | | Cantidad de pacientes que existen en la central de pacientes. | | Situación de error: | | | Si no existen pacientes debe lanzarse el error correspondiente. |  1. **[0.2]** Se necesita verificar si existen pacientes con GRIPA. Si no existen pacientes debe lanzarse el error correspondiente. Complete el requerimiento funcional.  |  |  |  |  | | --- | --- | --- | --- | | ID: | RF1 | Nombre: | verificar si existen pacientes con GRIPA | | Resumen: | | | Se necesita verificar si existen pacientes con GRIPA. Si no existen pacientes debe lanzarse el error correspondiente. | | Entradas: | | | N/A | | Resultado: | | | Se verifico si existen o no pacientes con gripa, | | Situación de error: | | | Si no existen pacientes debe lanzarse el error correspondiente. |  1. **[0.2]** Se necesita calcular y devolver la cantidad de pacientes mujeres que fueron atendidos por COVID. Si no existen pacientes debe lanzarse el error correspondiente. Complete el requerimiento funcional.  |  |  |  |  | | --- | --- | --- | --- | | ID: | RF1 | Nombre: | cantidad de pacientes mujeres que fueron atendidos por COVID. | | Resumen: | | | Se necesita calcular y devolver la cantidad de pacientes mujeres que fueron atendidos por COVID. Si no existen pacientes debe lanzarse el error correspondiente. | | Entradas: | | | N/A | | Resultado: | | | Cantidad de pacientes mujeres que fueron atendidos por COVID. | | Situación de error: | | | Si no existen pacientes debe lanzarse el error correspondiente. |  1. **[1.0]** Programe el método que resuelve el requerimiento funcional en la clase correspondiente. No olvide lanzar la situación de error 2. **[0.5]** Asocie la funcionalidad del anterior método al metodo1 para que informe al usuario la cantidad de mujeres pacientes con gripa 3. **[0.2]** Organice el código de los métodos de los numerales 5 y 6 de este laboratorio 4. **[0.5]** Documente de manera completa el método programado en el numeral 5 5. **[0.2]** Se necesita saber si en la central de pacientes existen o no pacientes de una clínica dada por el usuario. Si no existen pacientes en la central debe informarse del error. Complete el requerimiento funcional.  |  |  |  |  | | --- | --- | --- | --- | | ID: | RF2 | Nombre: | existen o no pacientes de una clínica dada por el usuario | | Resumen: | | | Se necesita saber si en la central de pacientes existen o no pacientes de una clínica dada por el usuario. Si no existen pacientes en la central debe informarse del error. | | Entradas: | | | Clinica (nClinicaUsuario) | | Resultado: | | | Si o no, dependiendo si hay pacientes asociados a la clínica dada por el usuario | | Situación de error: | | | Si no existen pacientes en la central debe informarse del error.  Si no se ingresa un nombre de clínica valido debe informase del error. |  1. **[1.0]** Programe el método que resuelve el requerimiento funcional en la clase correspondiente. No olvide lanzar la situación de error 2. **[0.5]** Asocie el método anterior con el metodo2 para que muestre al usuario si existen o no pacientes con COVID-19 3. **[0.2]** Organice el código de los métodos de los numerales 10 y 11 de este laboratorio 4. **[0.5]** Documente de manera completa el método programado en el numeral 10 5. **Seleccione uno de los problemas resuelto en este laboratorio y explíquelo mediante un video de no más de 10 minutos. Debe describir la manera en que programó tanto el método construido como su llamado en el metodo1 o metodo2. Suba el video a youtube, coloque en un archivo de texto el link y comprímalo junto con el ejercicio resuelto.** 6. Incluye este documento debidamente diligenciado en la carpeta docs/specs de la aplicación, comprime tu ejercicio y envíalo por la plataforma virtual. | | |