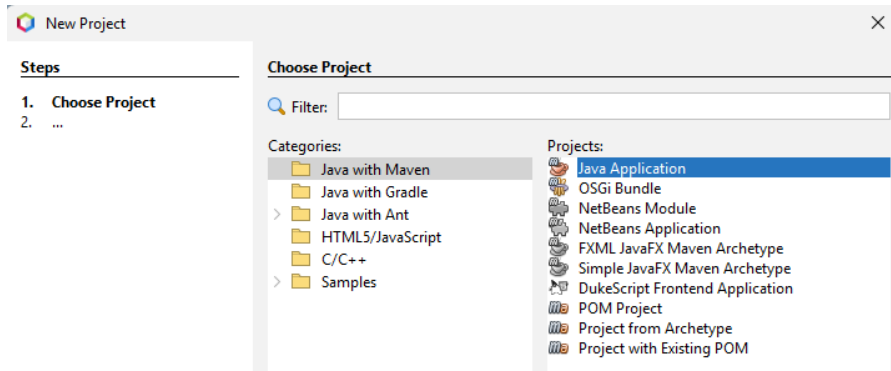


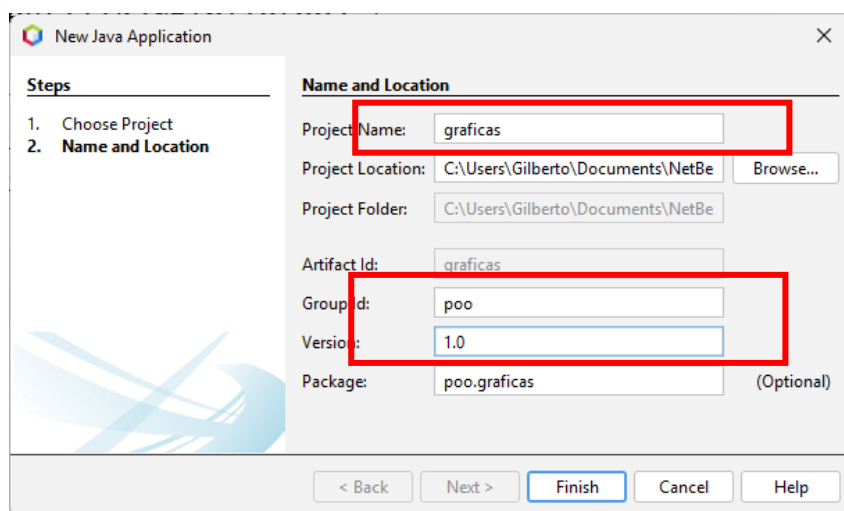
EJERCICIO EXTRA: 3. Creación de gráficas con JfreeChart

Objetivo. Aprender a crear instalar JfreeChart desde el Repositorio Central de Maven y crear un programa que grafique los datos.

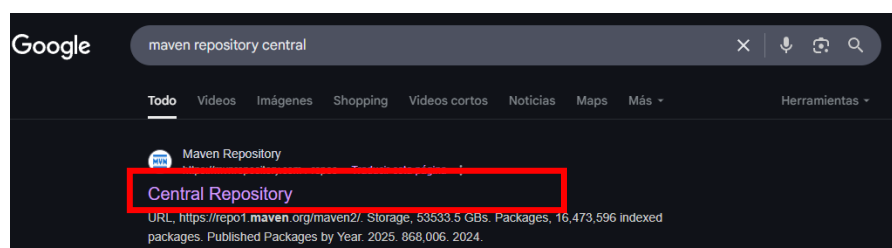
1. Abre Netbeans y elige un proyecto Java Application with Maven y damos clic en el botón **Next**.



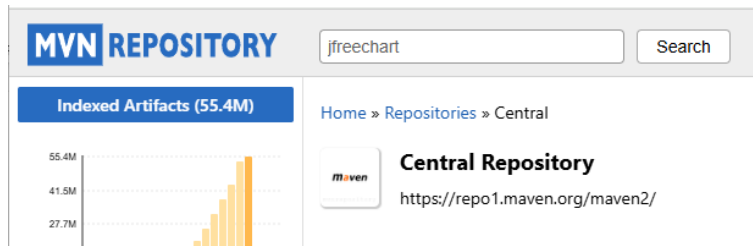
2. El nombre del proyecto será **graficas**, Group Id: **poo** y versión: **1.0**, tal como se muestra en la siguiente imagen y damos clic en el botón **Finish**.



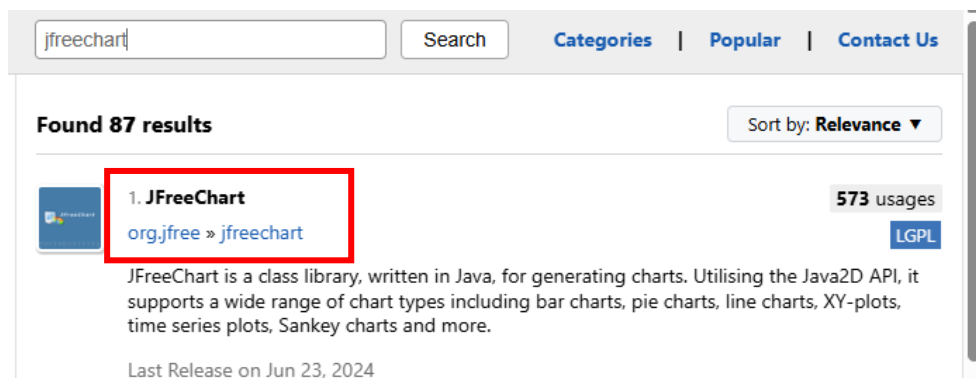
3. Abre el navegador de tu preferencia, escribe **Maven Repository Central** y da clic en la siguiente opción.



4. Una vez dentro de la página web escribe en la barra de búsqueda **jfreechart** y da clic en el botón **Search**.



5. Elige el primer resultado que aparezca



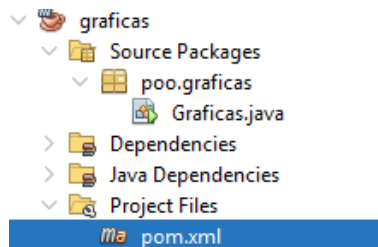
6. Elige la versión más actual (1.5.5) liberada en Junio del 2024.

Central (11) Atlassian (3) Redhat GA (4) FenixEdu (1) Mars-Sim (2)				
NumericalMethod (1) Talan (1) ICM (2)				
Version	Vulnerabilities	Repository	Usages	Date
1.5.5		Central	32	Jun 23, 2024
1.5.4		Central	69	Jan 08, 2023
1.5.3		Central	65	Feb 21, 2021
1.5.2		Central	7	Dec 31, 2020

7. Asegúrate de que esté seleccionada la pestaña de Maven, selecciona y copia el código que ahí aparece.



8. Abrimos el archivo pom.xml de nuestro proyecto dando doble clic sobre él.



9. Debajo de `</properties>` agregamos las siguientes etiquetas.

```
</properties>
<dependencies>

</dependencies>
</project>
```

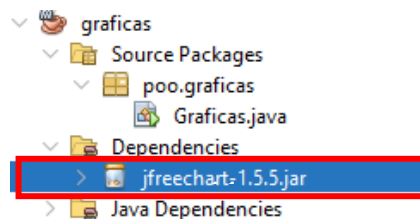
10. Debajo de la etiqueta **<dependencies>** pegamos el código copiado del repositorio central de maven. El código debe tener la siguiente apariencia.

```
</properties>
<dependencies>
  <!-- https://mvnrepository.com/artifact/org.jfree/jfreechart -->
  <dependency>
    <groupId>org.jfree</groupId>
    <artifactId>jfreechart</artifactId>
    <version>1.5.5</version>
  </dependency>
</dependencies>
```

Lo que acabamos de hacer es indicarle a nuestro proyecto actual que vamos a utilizar una dependencia (recurso) con los valores especificados misma que se descargará e instalará desde el repositorio central de maven.

11. Damos clic en el botón **clean and build project**  y esperamos a que se descargue el **jfreechart** desde internet

NOTA. Observemos que dentro de la carpeta **Dependencies** encontraremos al **jfreechart**, que ahora podemos utilizar.



12. Elimina todos los comentarios de la clase Graficas y agrega los siguientes imports de bajo del package.

```
import javax.swing.JFrame;
import javax.swing.WindowConstants;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.data.general.DefaultPieDataset;
```

13. Agregue el siguiente método a la clase para crear una gráfica de barras.

```
public static void graficaBarras(){
    DefaultCategoryDataset datos = new DefaultCategoryDataset();
    datos.setValue(98, "Programación II", "Luis");
    datos.setValue(90, "Programación II", "Laura");
    datos.setValue(78, "Programación II", "Karen");
    datos.setValue(95, "Programación II", "Diego");
    JFreeChart grafica;
    grafica = ChartFactory.createBarChart("Promedios", "Alumnos", "Calificacion", datos);
    ChartPanel panel = new ChartPanel(grafica);

    JFrame ventana = new JFrame();
    ventana.add(panel);
    ventana.setSize(800, 600);
    ventana.setVisible(true);
    ventana.setLocationRelativeTo(null);
    ventana.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
}
```

14. Agregue el siguiente método a la clase para crear una gráfica de circular.

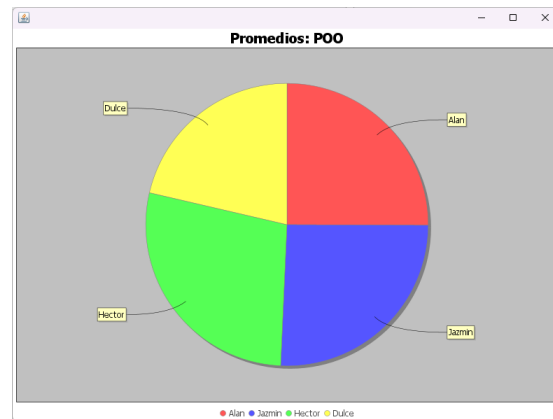
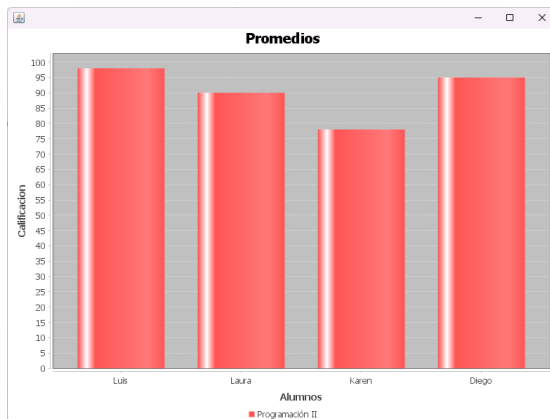
```
public static void graficaCircular(){
    DefaultPieDataset datos = new DefaultPieDataset();
    datos.setValue("Alan", 88);
    datos.setValue("Jazmin", 90);
    datos.setValue("Hector", 98);
    datos.setValue("Dulce", 75);
    JFreeChart grafica;
    grafica = ChartFactory.createPieChart("Promedios: POO", datos, true, true, false);
    ChartPanel panel = new ChartPanel(grafica);

    JFrame ventana = new JFrame();
    ventana.add(panel);
    ventana.setSize(800, 600);
    ventana.setVisible(true);
    ventana.setLocationRelativeTo(null);
    ventana.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
}
```

15. Agregue la llamada a los métodos.

```
public class Graficas {
    public static void main(String[] args) {
        graficaBarras();
        graficaCircular();
    }
}
```

16. Ejecute la aplicación y debido a que las ventanas tendrán el mismo tamaño y ubicación será necesario moverlas para poderlas ver juntas.



Actividad para realizar:

Crea 2 métodos llamados: **graficaLinea** y **graficaAnillos**, donde crees una gráfica de tipo **lineChart** y **ringChart**, respectivamente.