



CARRERA DESARROLLO DE SOFTWARE

PROGRAMACION ORIENTADA A OBJETOS

TAREA: MINI CONCURSO DE NATACION

MARCELA UMAGINGA

marcela.umaginga@cenestur.edu.ec

YADIRA FRANCO

Quito, Ecuador 2025

- **PREGUNTAS DE OPCION MULTIPLE: CONCURSO DE NATACION.**

¿Qué devuelve el método puedeParticipar() cuando el participante tiene una condición de salud "mala"?

- a) verdadero
- b) falso
- c) nulo
- d) Un mensaje de error

Respuesta correcta: b) falso

¿Qué relación describe mejor la conexión entre la clase Participante y su atributo de salud?

- a) Agregación
- b) Composición
- c) Herencia
- d) Asociación

Respuesta correcta: d) Asociación

En POO, ¿qué significa que una clase sea abstracta?

- a) Que tiene al menos un método con implementación básica
- b) Que no se puede instanciar directamente
- c) Que puede contener solo variables estáticas
- d) Que hereda de todas las subclases

Respuesta correcta: b) Que no se puede instanciar directamente

¿Qué sucede al sobrescribir un método en una subclase?

- a) Se crea un método nuevo sin relación al original
- b) El método original se elimina
- c) Se modifica el comportamiento para que sea específico en la subclase

d) El método sobrescrito no puede ser llamado nunca

Respuesta correcta: c) Se modifica el comportamiento para que sea específico en la subclase

¿Cuál es un rango típico válido para la edad de un participante en el sistema?

a) 15 a 30

b) 18 a 25

c) 20 a 35

d) Cualquier edad es válida

Respuesta correcta: b) 18 a 25

¿Qué nivel de acceso permite que los atributos sean visibles solo dentro de su clase y sus subclases?

a) público

b) privado

c) protegido

d) estático

Respuesta correcta: c) protegido

¿Qué patrón de diseño podría mejorar la separación entre la interfaz de usuario y la lógica del sistema?

a) Singleton

b) MVC (Modelo-Vista-Controlador)

c) Iterador

d) Fábrica

Respuesta correcta: b) MVC (Modelo-Vista-Controlador)

¿Cómo se podría almacenar una gran cantidad de participantes de manera dinámica durante la ejecución?

a) En un array con tamaño fijo

- b) En una lista dinámica como ArrayList
- c) En variables individuales
- d) Solo en un archivo

Respuesta correcta: b) En una lista dinámica como ArrayList

¿Cuál es el tipo de dato más adecuado para validar si un participante puede participar o no?

- a) Cadena
- b) booleana
- c) int
- d) char

Respuesta correcta: b) booleano

¿Qué atributo extra tendría una clase ParticipanteUniversidad en comparación con Participante?

- a) nombreUniversidad
- b) tipoSindicato
- c) códigoEmpleado
- d) nivelEstudios

Respuesta correcta: a) nombreUniversidad

Para crear un menú en consola que repita opciones hasta que el usuario decida salir, ¿qué estructura es apropiada?

- a) si-si no
- b) para
- c) hacer-mientras
- d) cambiar sin ciclo

Respuesta correcta: c) hacer-mientras

¿Para qué sirve llamar scanner.nextLine() después de scanner.nextInt()?

- a) Para leer y descartar el salto de línea pendiente en el buffer

- b) Para cerrar el escáner
- c) Para leer otro número entero
- d) Para lanzar una excepción

Respuesta correcta: a) Para leer y descartar el salto de línea pendiente en el buffer

¿Qué debe mostrar un método getInfo() bien implementado en esta aplicación?

- a) El resultado del método puedeParticipar()
- b) Información completa y formateada sobre el participante
- c) El resultado del registro en pantalla
- d) El estado del menú

Respuesta correcta: b) Información completa y formateada sobre el participante

¿Qué debería hacer el sistema si el participante tiene 26 años y el límite es 25?

- a) Ajustar la edad a 25 automáticamente
- b) Permitir el registro igualmente
- c) Denegar el registro y mostrar un mensaje de error
- d) Ignorar la edad y continuar

Respuesta correcta: c) Denegar el registro y mostrar un mensaje de error

¿Cuál es un tipo de dato adecuado para manejar un atributo que indica si un participante está saludable?

- a) booleano
- b) int
- c) cadena
- d) flotante

Respuesta correcta: a) booleano

¿Cuál de las siguientes clases no es probable que herede de Participante?

- a) ParticipanteColegio
- b) ParticipanteProfesor
- c) ParticipanteEmpresa
- d) ParticipanteEvento

Respuesta correcta: d) ParticipanteEvento

¿Cómo debería actuar el sistema frente a un participante que no cumple todos los requisitos?

- a) Registrar al participante y permitir competir
- b) Rechazar el registro sin almacenarlo
- c) Modificar automáticamente los datos para que cumpla
- d) Registrar pero marcarlo como no elegible

Respuesta correcta: b) Rechazar el registro sin almacenarlo

¿Qué información proporciona generalmente getInfo() en un ParticipanteSindicato?

- a) Solo el nombre y edad
- b) Todos los atributos incluyendo sindicato al que pertenece
- c) Solo el nombre del sindicato
- d) Mensaje de error si no es sindicato

Respuesta correcta: b) Todos los atributos incluyendo sindicato al que pertenece

¿Qué modificador se utiliza para proteger atributos y métodos para que sean accesibles solo dentro de la clase y subclases?

- a) privado
- b) protegido
- c) público
- d) predeterminado

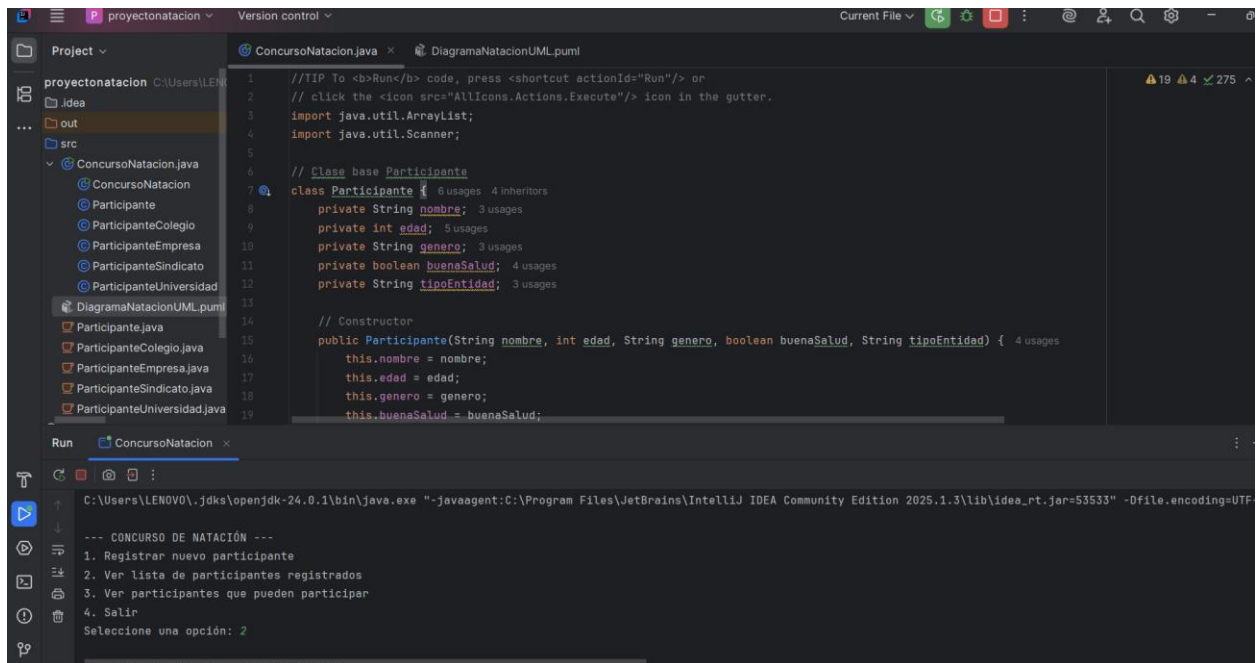
Respuesta correcta: b) protegido

¿Por qué es útil usar el polimorfismo en métodos como getInfo() dentro de clases derivadas?

- a) Para compartir código sin cambios
- b) Para permitir diseños estáticos y rígidos
- c) Para que cada clase muestre información específica reusando la estructura común
- d) Para impedir que se modifiquen datos

Respuesta correcta: c) Para que cada clase muestre información específica reutilizando la estructura común

CODIGOS CONCURSO DE NATACION



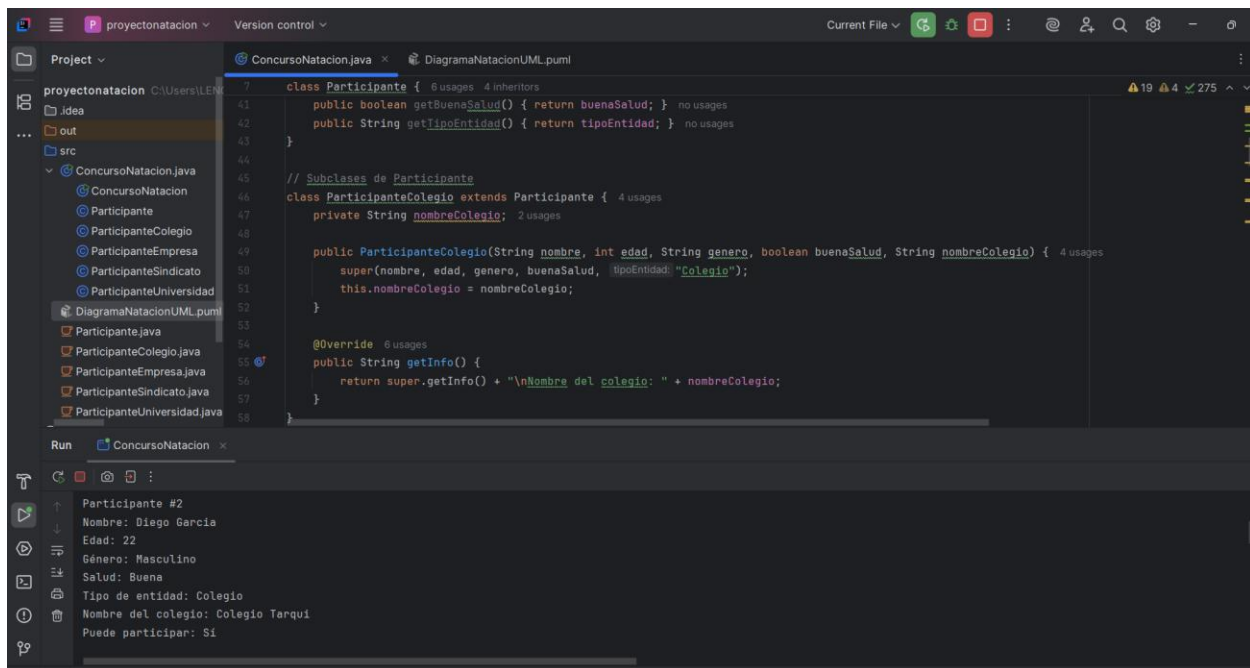
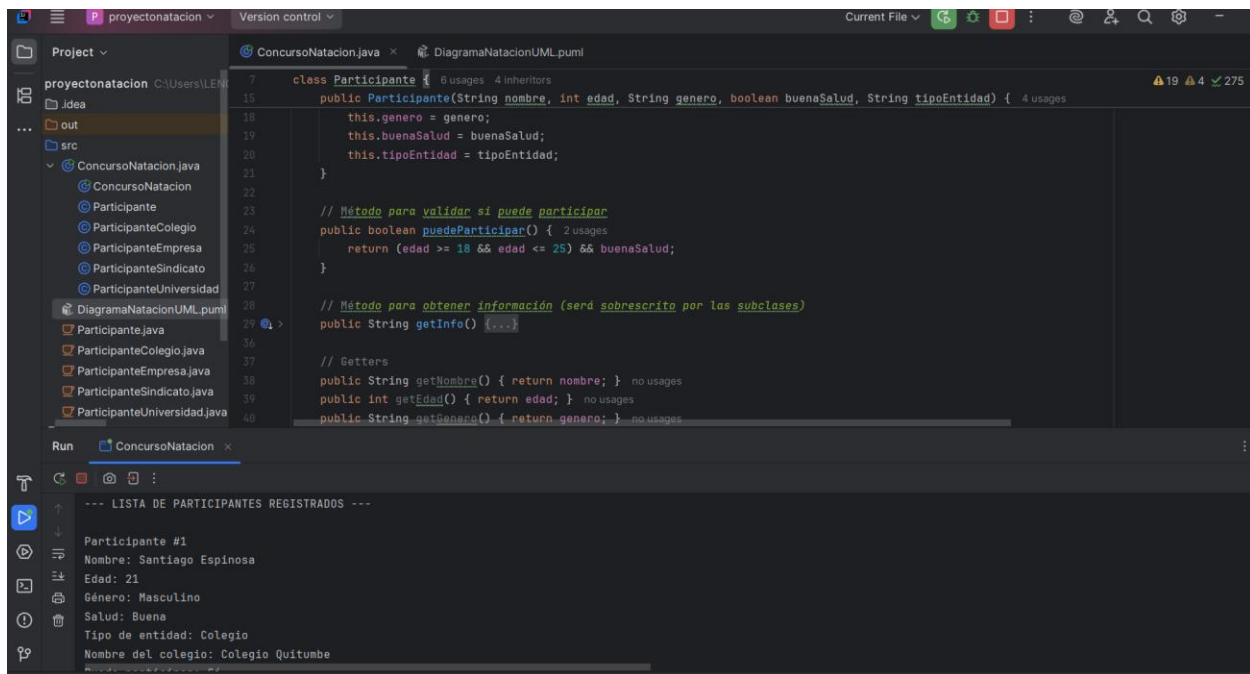
```
1 //TIP To <b>Run</b> code, press <shortcut actionId="Run"/> or
2 // click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.
3 import java.util.ArrayList;
4 import java.util.Scanner;
5
6 // Clase base Participante
7 class Participante {
8     private String nombre;
9     private int edad;
10    private String genero;
11    private boolean buenaSalud;
12    private String tipoEntidad;
13
14    // Constructor
15    public Participante(String nombre, int edad, String genero, boolean buenaSalud, String tipoEntidad) {
16        this.nombre = nombre;
17        this.edad = edad;
18        this.genero = genero;
19        this.buenaSalud = buenaSalud;
20    }
21 }
```

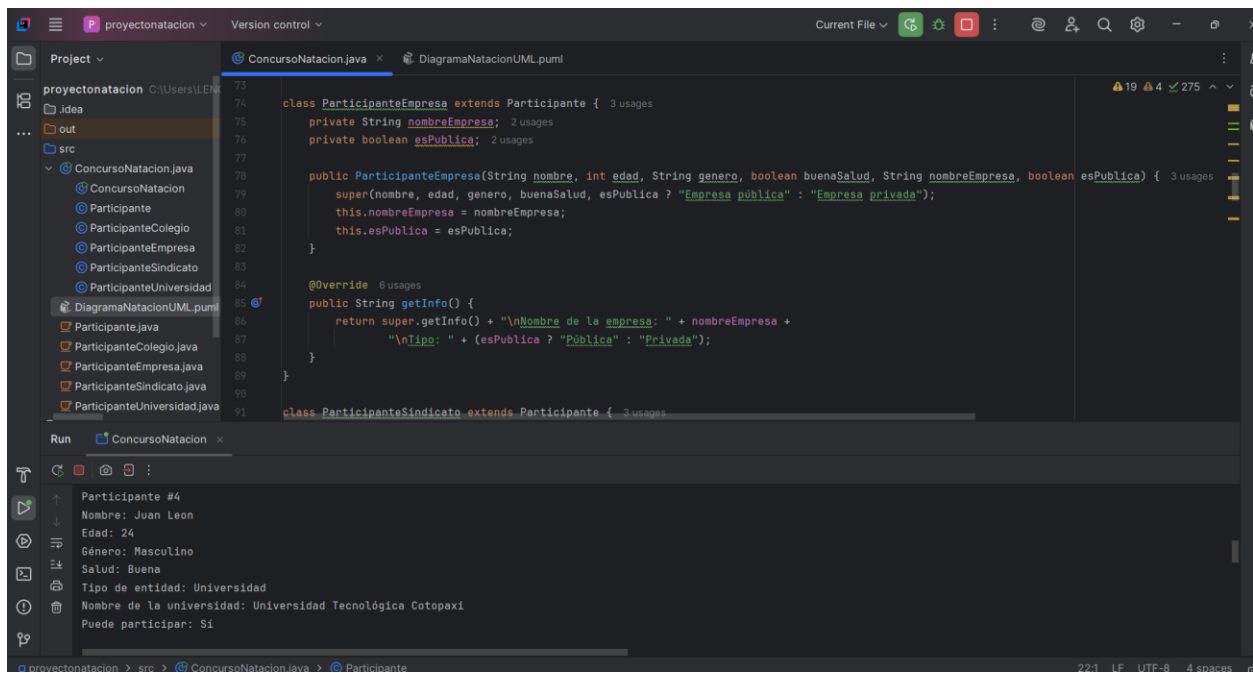
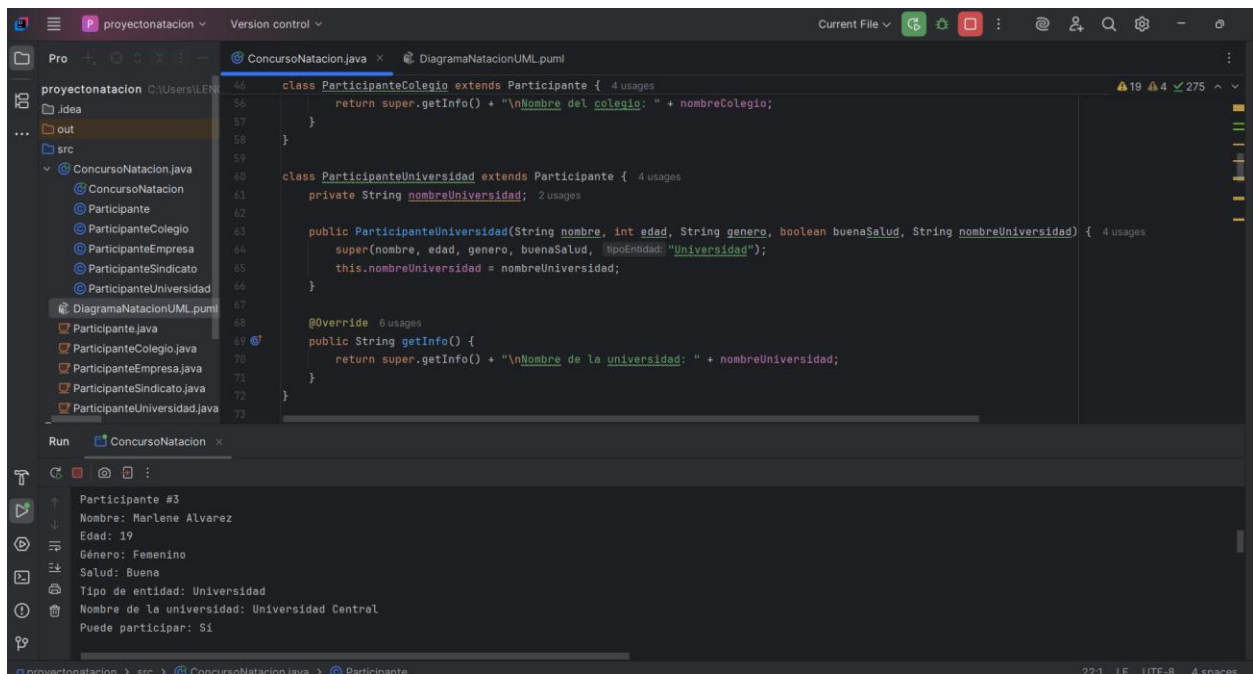
Project Structure:

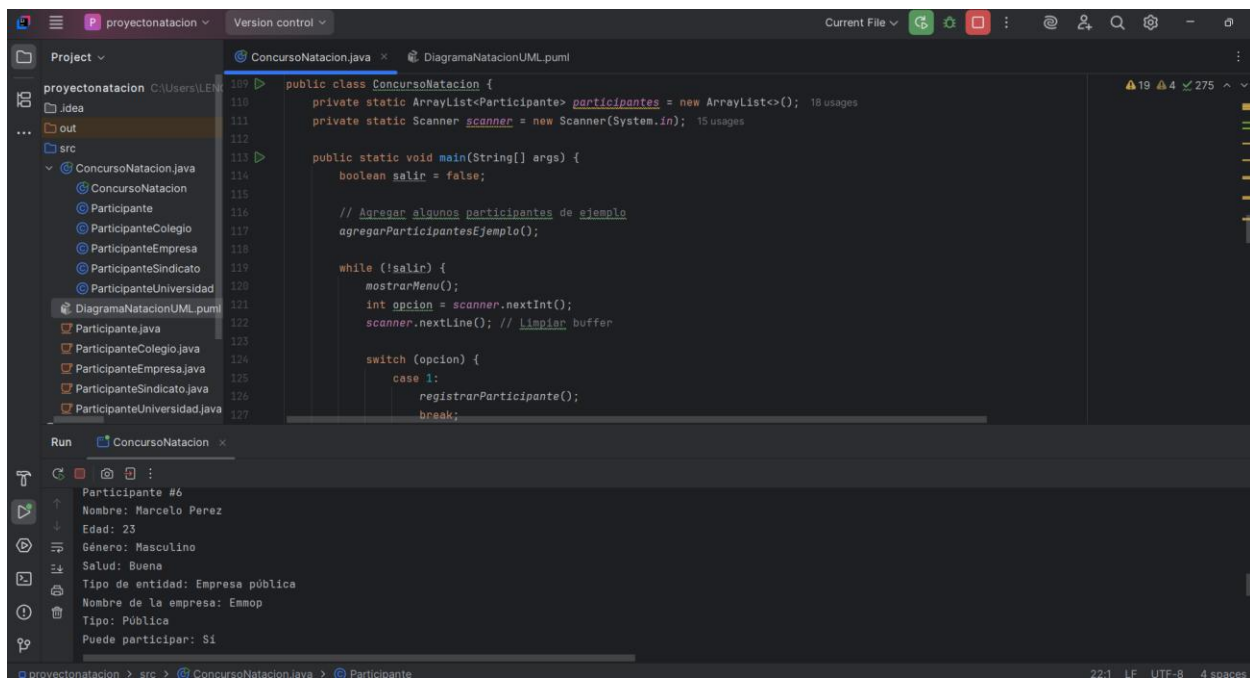
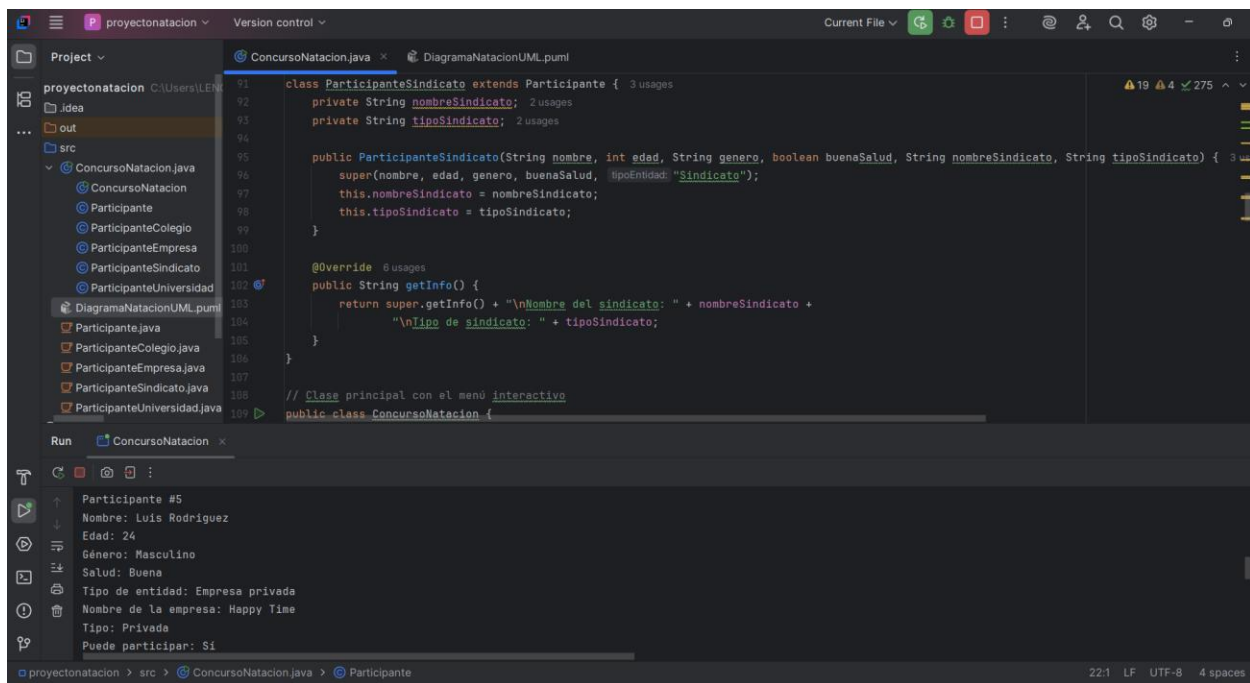
- projectonatacion
 - out
 - src
 - ConcursoNatacion.java
 - ConcursoNatacion
 - Participante
 - ParticipanteColegio
 - ParticipanteEmpresa
 - ParticipanteSindicato
 - ParticipanteUniversidad
 - DiagramaNatacionUML.puml
 - Participante.java
 - ParticipanteColegio.java
 - ParticipanteEmpresa.java
 - ParticipanteSindicato.java
 - ParticipanteUniversidad.java

Run Console Output:

```
C:\Users\LENDVO\jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.3\lib\idea_rt.jar=53533" -Dfile.encoding=UTF-8
--- CONCURSO DE NATACIÓN ---
1. Registrar nuevo participante
2. Ver lista de participantes registrados
3. Ver participantes que pueden participar
4. Salir
Seleccione una opción: 2
```







```
109 public class ConcursosNatacion {
110     public static void main(String[] args) {
111
112         switch (opcion) {
113             case 1:
114                 registrarParticipante();
115                 break;
116             case 2:
117                 listarParticipantes();
118                 break;
119             case 3:
120                 listarParticipantesElegibles();
121                 break;
122             case 4:
123                 salir = true;
124                 System.out.println("Saliendo del sistema...");
125                 break;
126             default:
127                 System.out.println("Opción no válida. Intente nuevamente.");
128         }
129     }
130 }
```

Run ConcursosNatacion

Participante #7
Nombre: Ivonne Castro
Edad: 19
Género: Femenino
Salud: Buena
Tipo de entidad: Sindicato
Nombre del sindicato: Sindicato de Vendedores
Tipo de sindicato: Comercio
Puede participar: Si

```
109 public class ConcursosNatacion {
110     private static void mostrarMenu() { 1 usage
111         System.out.println("\n--- CONCURSO DE NATACIÓN ---");
112         System.out.println("1. Registrar nuevo participante");
113         System.out.println("2. Ver lista de participantes registrados");
114         System.out.println("3. Ver participantes que pueden participar");
115         System.out.println("4. Salir");
116         System.out.print("Seleccione una opción: ");
117     }
118
119     private static void registrarParticipante() { 1 usage
120         System.out.println("\n--- REGISTRAR NUEVO PARTICIPANTE ---");
121
122         // Datos comunes
123         System.out.print("Nombre: ");
124         String nombre = scanner.nextLine();
125
126         System.out.print("Edad: ");
127     }
128 }
```

Run ConcursosNatacion

Participante #8
Nombre: Diego Herrera
Edad: 25
Género: Masculino
Salud: Buena
Tipo de entidad: Sindicato
Nombre del sindicato: Sindicato de Transportista
Tipo de sindicato: profesion
Puede participar: Si

The screenshot shows an IDE with the project 'proyeccionatacion'. The file explorer on the left shows the project structure, including 'ConcursoNatacion.java' and 'DiagramaNatacionUML.puml'. The main editor displays the code for 'ConcursoNatacion.java', specifically the 'registrarParticipante()' method. The code prompts the user for age, gender, and health status, then validates if the participant meets the requirements (age between 18 and 25, and good health). The Run window at the bottom shows the output for 'Participante #9', Emily Solorzano, a 17-year-old female with good health, who is a high school student and cannot participate.

```
109 public class ConcursoNatacion {
153     private static void registrarParticipante() {
159
160         System.out.print("Edad: ");
161         int edad = scanner.nextInt();
162         scanner.nextLine(); // Limpiar buffer
163
164         System.out.print("Género: ");
165         String genero = scanner.nextLine();
166
167         System.out.print("¿Tiene buena condición de salud? (si/no): ");
168         boolean buenaSalud = scanner.nextLine().equalsIgnoreCase("si");
169
170         // Validar requisitos básicos
171         if (edad < 18 || edad > 25 || !buenaSalud) {
172             System.out.println("\nEste participante no cumple con los requisitos básicos!");
173             System.out.println("Debe tener entre 18 y 25 años y estar en buena condición de salud.");
174             return;
175         }
176     }
177 }
```

Run: ConcursoNatacion

Participante #9
Nombre: Emily Solorzano
Edad: 17
Género: Femenino
Salud: Buena
Tipo de entidad: Colegio
Nombre del colegio: Colegio La Salle
Puede participar: No

The screenshot shows the same IDE with the 'ConcursoNatacion.java' file. The code continues from the previous snippet, showing the logic for selecting the entity type (Colegio, Universidad, Empresa, Sindicato) and creating a 'Participante' object based on the selection. The Run window shows the output for 'Participante #10', Marco Montero, a 20-year-old male with no good health, who is a university student and cannot participate.

```
177 // Tipo de entidad
178 System.out.println("\nTipo de entidad:");
179 System.out.println("1. Colegio");
180 System.out.println("2. Universidad");
181 System.out.println("3. Empresa");
182 System.out.println("4. Sindicato");
183 System.out.print("Seleccione el tipo de entidad: ");
184 int tipo = scanner.nextInt();
185 scanner.nextLine(); // Limpiar buffer
186
187 Participante participante = null;
188
189 switch (tipo) {
190     case 1:
191         System.out.print("Nombre del colegio: ");
192         String nombreColegio = scanner.nextLine();
193         participante = new ParticipanteColegio(nombre, edad, genero, buenaSalud, nombreColegio);
194     }
195 }
```

Run: ConcursoNatacion

Participante #10
Nombre: Marco Montero
Edad: 20
Género: Masculino
Salud: No buena
Tipo de entidad: Universidad
Nombre de la universidad: Universidad Cotopaxi
Puede participar: No

DIAGRAMA

