

Pautas:

- Se recomienda conectarte a la meet con tu cuenta de la UTN.
- No intentes conectarte antes de tu turno, respetá el horario, puede haber otra persona rindiendo en la misma sala virtual.
- Se te pedirá que tengas la cámara prendida todo el tiempo.
- El final será grabado.
- Al iniciar el examen, deberás mostrar tu DNI a la cámara y decir tu nombre, tu DNI y tu legajo.
- Se tomarán los 3 paradigmas: objetos, funcional y lógico, y también se podrán hacer preguntas de integración/comparación entre ellos.
- Entre otras cosas, se te mostrará código y se harán preguntas sobre ese código que responderás oralmente. Quienes estén tomando el final te harán las repreguntas necesarias para asegurarse de que sabés el tema.
- El final tendrá una duración aproximada de 20 minutos.
- Luego, el equipo docente se tomará unos minutos para decidir si aprobaste (o no) y tu nota, y te la informarán en el momento.

Para aprobar, deberás demostrar conocimiento **en los tres paradigmas**, y además deberás responder las preguntas de **integración** correctamente. Esto significa que si no demostrás conocimiento suficiente en alguno de los paradigmas, no podés aprobar.

Objetos

Suponiendo que está implementado todo lo necesario para que este código ande:

```
class Socio {
  const compras

  method saldo() =
    compras.sum({ compra => compra.precio() } )
}

class SocioVIP inherits Socio {
  override method saldo() =
    compras.sum({ compra => compra.precio() } ) * 0.8
}
```

Preguntas:

- ¿Qué hace el código?
- ¿Se está aprovechando la herencia?
- ¿Es importante que el método de SocioVIP también se llame saldo()?

Funcional

Dadas las funciones:

```
f1 h = any (> 15) . map h
```

```
f2 h = (> 0) . length . filter (> 15) . map h
```

Preguntas:

- ¿Qué hacen f1 y f2?
- ¿Cuál es la diferencia entre ambas?
- ¿De qué tipo debería ser el parámetro h?

Lógico

```
tiene(juan, auto).
```

```
tiene(juan, moto).
```

```
tiene(juan, cuatri).
```

```
tiene(pedro, moto).
```

```
tiene(vanesa, tractor).
```

```
cuesta(auto, 15).
```

```
cuesta(moto, 10).
```

```
cuesta(cuatri, 13).
```

```
cuesta(tractor, 18).
```

% Alguien es fan de lo barato si tiene todos los vehículos baratos que existen (es barato si cuesta 15 o menos)

```
fanDeLoBarato(Persona):-
```

```
    forall(tiene(Persona, Vehiculo), (cuesta(Vehiculo, Costo), Costo <= 15)).
```

?- fanDeLoBarato(Quien).

false

Preguntas:

- ¿Funciona de acuerdo a lo pedido? ¿Por qué?
- ¿Hay problemas de inversibilidad? ¿Por qué?