

Bienvenido a tu Certificación

Ya estás un paso más cerca de ser un Analista del Conocimiento - Dimensión Programador!

El examen consta de 6 ejercicios basados en los conocimientos exigidos por el 111 Mil y por la industria, para convertirte en programador junior. Tené en cuenta y leé con mucha atención las siguientes pautas para la correcta realización y aprobación del examen:

- 1- El examen tiene una duración máxima de 3 (tres) horas. Deberás enviar los resultados antes de cumplir ese tiempo.
 - 2- Es necesario responder TODOS los ejercicios para poder aprobar el examen.
 - 3- El resultado es APROBADO o DESAPROBADO, no tiene puntaje.
 - 4- Los veedores estarán presentes para verificar que el examen se tome en las condiciones adecuadas.
 - 5- Está prohibido utilizar el chat, el correo electrónico (fuera del uso normal para abrir este formulario), o cualquier página web que no sea este Google Form.
 - 6- Tené presente que el teclado no reconoce la tecla "TAB" para escribir código Java, por lo que deberás usar 3 (tres) veces la tecla "ESPACIO" para poner sangrías (indentar/tabular) al alinear el código.
 - 7- En caso de no encontrar el tipeo de una tecla, podrás usar el mapa de caracteres, que podés encontrarlo en: "Tecla Windows + R" y en el cuadro de texto escribir "charmap", y apretar "Enter". También podés encontrarlo en "Inicio--> Accesorios ---> Herramientas del Sistema ---> Mapa de caracteres".
 - 8- Cuando en el punto siguiente el formulario te pida el código de seguridad, tenés que solicitárselo a tu veedor.
- Te deseamos mucha suerte y a trabajar en los ejercicios!

***Obligatorio**

Datos Personales

1. DNI (sin puntos) *

2. Apellidos *

3. Nombres *

4. Fecha de Nacimiento *

Ejemplo: 15 de diciembre de 2012

5. Dirección de correo electrónico *

6. Situación de Examen **Marca solo un óvalo.*

- ☐ SIN CURSADA - LIBRE
- ☐ Realicé el curso del Plan 111Mil

7. Sede de Certificación **Marca solo un óvalo.*

- ☐ C.F.P. N°16
- ☐ CEDER Río Tercero
- ☐ Centro Regional Universitario de Bragado
- ☐ Direccion de Empleo
- ☐ E.E.T. N° 3139 'Gral. Martín Miguel de Güemes'
- ☐ EET N°3109 "OEA"
- ☐ ESCUELA DE EDUCACION TECNICA N° 3104 LANZA COLOMBRES EX N°5104-7105
- ☐ ESCUELA DE EDUCACION TECNICA N° 3132 EX N° 5132
- ☐ Escuela Santo Domingo
- ☐ Escuela Técnica N° 1 de Concepción
- ☐ Escuela Técnica N° 20 "Antonio Berni"
- ☐ IDI Facultad de Tecnología y Ciencias Aplicadas - UNCA
- ☐ Instituto CyP Soft
- ☐ Instituto Tecnológico Goya
- ☐ Ministerio de Educación Sala Blanca
- ☐ Ministerio de Produccion y Trabajo de la Nacion
- ☐ Polo Tecnologico de Parana
- ☐ Punto Digital Escuela de Arte y Oficios
- ☐ Punto Digital Pacto Social
- ☐ Universidad Nacional de La Rioja
- ☐ Universidad Nacional de San Juan
- ☐ Universidad Nacional de San Martin - Sede Caseros
- ☐ UNIVERSIDAD NACIONAL DEL LITORAL
- ☐ Universidad Provincial de Ezeiza
- ☐ UNN - Fac de Ciencias Exactas y Naturales y Agrimensura
- ☐ UTN - Mar del Plata
- ☐ UTN - Regional Resistencia
- ☐ UTN - Regional Tucuman
- ☐ UTN - REGIONAL VILLA MARIA
- ☐ UTN Facultad Regional Córdoba.
- ☐ UTN Facultad Regional San Francisco
- ☐ UTN FR Rosario

8. Solicitud de Certificado de Asistencia *

En caso de necesitar un certificado que indique que estuviste presente en el examen para tu trabajo u otra organización indícalo aquí. El mismo te llegará en forma digital, al correo electrónico que indicaste al comienzo de este formulario
Marca solo un óvalo.

- ☐ Sí
- ☐ No

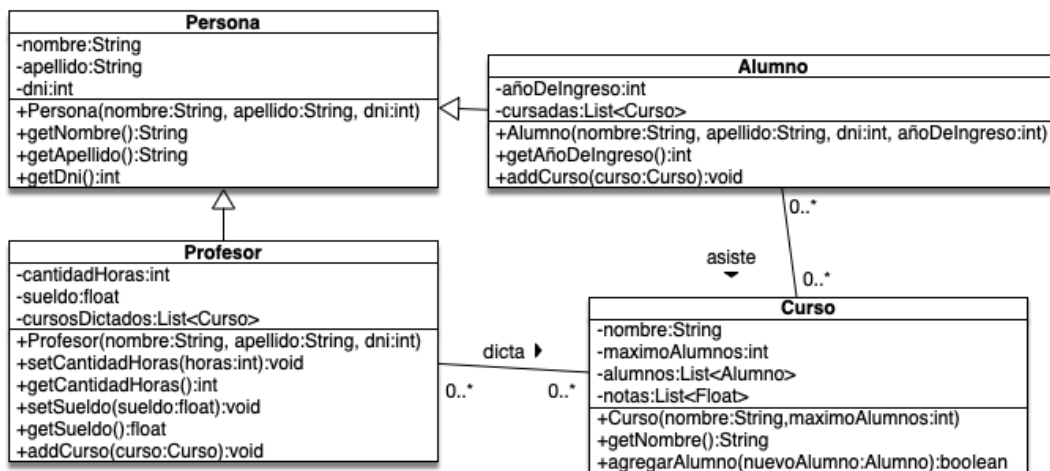
Contexto del examen - Cursos

La Universidad Nacional de Westeros (UNW) decidió comenzar a desarrollar un sistema para administrar los cursos que se dictan cada año. La aplicación permitirá registrar los cursos que se ofrecen, junto a los alumnos inscriptos en los mismos y los profesores que los dictan. Como saben que cursamos el programa 111Mil nos invitaron a ser parte del desarrollo.

Ejercicio 1. Implementar desde el diagrama de clases

Docentes de la UNW realizaron un diagrama UML preliminar del sistema y nos pidieron que implementemos la clase Alumno según el diagrama (el resto de las clases serán implementadas por la UNW). Inicialice los atributos que crea necesarios. Los docentes nos indicaron que, una vez que hayamos implementado la clase Alumno, ejecutarán el siguiente código para comprobar que no existen errores:

```
Curso curso111= new Curso("111", 40);
Alumno alumno=new Alumno("Joffrey", " Baratheon", 25346123, 2018);
alumno.addCurso(curso111);
```

**9. Implemente la clase Alumno**

Ejercicio 2. Implementar un método a partir de un enunciado

Programar en Java la funcionalidad para obtener todos los cursos que dicta un determinado profesor que tengan al menos un alumno que haya ingresado con anterioridad a un determinado año pasado por parámetro.

Implemente los métodos que considere necesarios indicando para cada uno de ellos a qué clase corresponde.

10. Indique en qué clase/s corresponde dicha funcionalidad e implemente el/los método/s necesarios para satisfacer este nuevo requerimiento

Ejercicio 3. Implementar y documentar

Dada la gran aceptación que tienen los cursos de la UNW, cada curso tiene un número máximo de alumnos que pueden inscribirse. Este número varía según el curso. Dado que los docentes de la UNW están muy ocupados preparando exámenes para sus alumnos, no pudieron terminar la implementación de esta funcionalidad en la clase Curso. Por ello, nos solicitaron que implementemos el método agregarAlumno según se indica en el diagrama de clases. El mismo recibe por parámetro el alumno a inscribir. El método solo debe agregar al alumno a la lista alumnos si aun no se supero el número máximo del curso. En caso de cumplirse esta condición, el método además deberá agregar el curso en la lista cursos del alumno inscripto y retornar true. En caso contrario retornará false.

Implemente el método solicitado y elabore la documentación técnica utilizando Javadoc. Incluya tanto la descripción del método como los tags que correspondan.

11. Implementación y Documentación en Javadoc

Ejercicio 4. Seguimiento de código

Los docentes de la UNW sobrescribieron el método toString en las clases Persona y Alumno de la siguiente manera:

Clase Persona

```
@Override
public String toString() {
    return this.nombre+" "+ this.apellido+" "+ this.dni;
}
```

Clase Alumno

```
@Override
public String toString() {
    return super.toString()+" "+ this.añoDelIngreso+" "+ this.cursadas.size();
}
```

Considerando estos cambios, ¿Qué imprimirá el programa al ejecutar el siguiente código? Recuerde

que el método agregarAlumno inscribe a un alumno en un Curso siempre y cuando no se haya alcanzado el número máximo de inscriptos.

```

Curso c1=new Curso("Programacion", 4);
Curso c2=new Curso("Bases de Datos", 2);
Curso c3=new Curso("Tecnicas", 3);

Alumno a1=new Alumno("Daenerys", "Targaryen", 123456, 2018);
Alumno a2=new Alumno("Jon", "Snow", 123457, 2017);
Alumno a3=new Alumno("Cersei", "Lannister", 123458, 2018);

c1.agregarAlumno(a1);
c1.agregarAlumno(a2);
c1.agregarAlumno(a3);

c2.agregarAlumno(a1);
c2.agregarAlumno(a2);
c2.agregarAlumno(a3);

c3.agregarAlumno(a2);

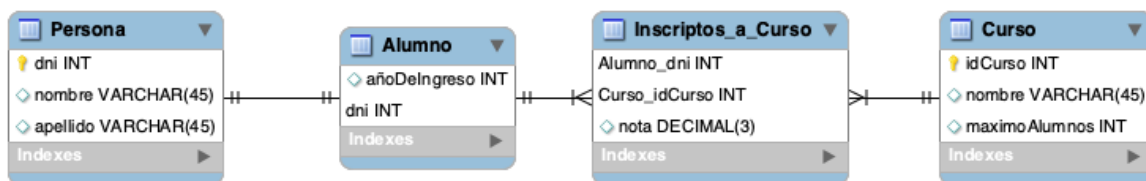
System.out.println(a1);
System.out.println(a2);
System.out.println(a3);

```

12. Qué imprime?

Ejercicio 5. Consulta SQL

Se ha modelado el siguiente diagrama de Entidad Relación.



Consulta SQL

Dado el diagrama de entidad relación, escriba la consulta SQL que liste la cantidad de alumnos ingresantes por año de ingreso (del año mas reciente al mas antiguo). La consulta solo debe tener en cuenta los alumnos que están inscriptos en al menos un curso. Además, dadas las siguientes tuplas de ejemplo, determinar el resultado de la consulta.

Persona		
123456	Daenerys	Targaryen
123457	Jon	Snow
123458	Cersei	Lannister
Alumno		
2018	123456	
2017	123457	
2018	123458	

Inscriptos a curso		
123456	1	8
123457	1	5
123458	1	7
123456	2	7
123457	2	4
123458	2	5
123456	3	10
123457	3	6
123458	3	9
Curso		
1	Programa cion	30
2	Bases de datos	30
3	Tecnicas	40

13. SQL y Resultado de la consulta

Ejercicio 6. Hibernate

Dado el diagrama de entidad-relación presentado en el ejercicio anterior y el diagrama UML presentado en el ejercicio 1, escriba la línea del archivo de mapeo de Hibernate (en formato XML o anotación) correspondiente al mapeo del atributo “apellido”.

14. Hibernate

Código de Seguridad

Ingresá el código de seguridad correspondiente

15. Código de Seguridad *

Te recordamos revisar tu examen y tu conexión a internet antes de presionar el botón "ENVIAR", ya que una vez que lo hagas, tus respuestas serán enviadas y no podrás modificarlas.

Éxitos!

Con la tecnología de
 **Google Forms**