

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.

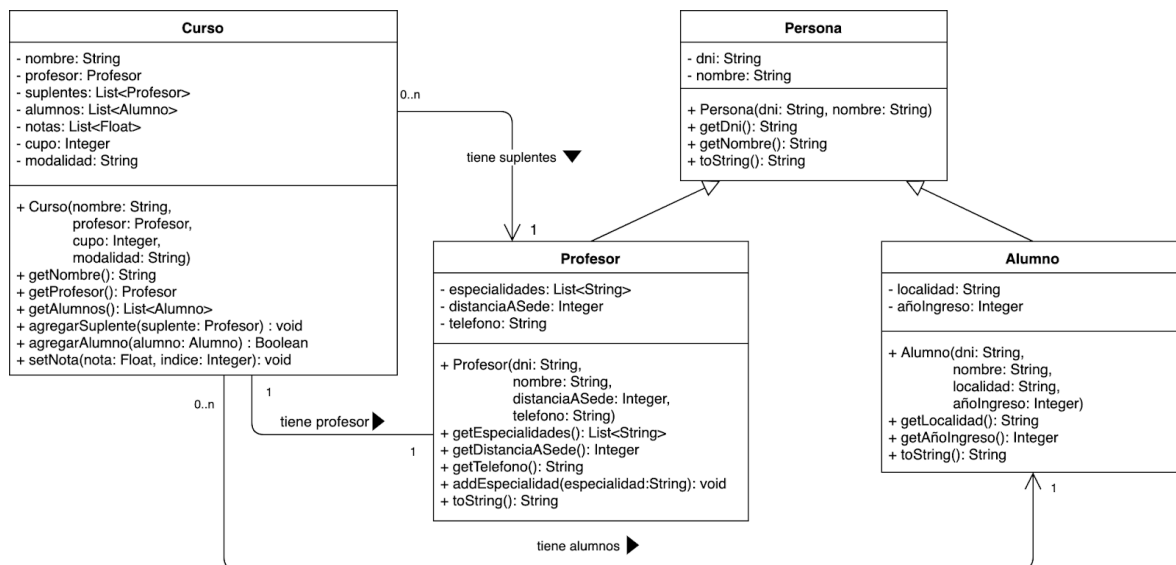
- ☐ Sí
- ☐ No

Contexto del examen - Cursos de Oficio

La Universidad Nacional de Misiones (UNAM) brinda, al igual que muchas otras universidades en el país, cursos de oficios abiertos a la comunidad. Sin embargo, no existe hoy en día un sistema interno para administrarlos. Por ello, a los alumnos de 111Mil se les ocurrió crear una aplicación Java y donarla a la UNAM.

Ejercicio 1. Implementar desde el diagrama de clases

En la primer etapa de desarrollo, se creó un diagrama UML con la estructura básica de las clases de la aplicación. Se le solicita a Ud. que implemente la clase Profesor a partir del diagrama. Tome en cuenta que el método toString retorna un String construido a partir de los siguientes campos: dni, nombre, teléfono, número de especialidades y distancia a la sede, separados por coma.
 Profesor profesor = new Profesor("1234", "Profesor Prueba", 50, "+54 223 123456789");
 profesor.addEspecialidad("teatro");
 System.out.println(profesor); // Debería imprimir: 1234, Profesor Prueba, +54 223 123456789, 1, 50



9. Implemente la clase Profesor

Ejercicio 2. Implementar un método a partir de un enunciado

En invierno es normal que un profesor anuncie, horas antes de comenzar el curso, que no podrá asistir por enfermedad. Se le pide a Ud. implementar un método llamado `getMejorSuplente` que, utilizando la lista de profesores suplentes, obtenga aquel Profesor que se encuentre a la menor distancia de la sede. Solo deben tomarse en cuenta Profesores cuyos teléfonos no sean un String vacío. En caso de que 2 profesores con teléfono estén a la misma distancia, elegir aquel que tenga más especialidades.

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

Implemente el método `agregarAlumno` de la clase `Curso`, el cual recibe un `Alumno` por parámetro y agrega el alumno a la lista de alumnos del curso.
Dicho método verifica que, si la modalidad del curso es "presencial", no se supere el cupo establecido y que la localidad del Alumno sea "Posadas". En caso de que la modalidad del curso sea "virtual", no hay cupo máximo ni verificación de localidad.
El método retorna `true` si se pudo agregar al alumno y `false` en caso contrario.

Implemente el método solicitado y elabore la documentación técnica utilizando Javadoc. Incluya tanto la descripción del método así como también las anotaciones que correspondan.

11. Implementación y Documentación en Javadoc

Ejercicio 4. Seguimiento de código

¿Qué imprimirá el programa al ejecutar el siguiente código? Recuerde que el método `agregarAlumno` verifica la modalidad del curso y el cupo del mismo (Ejercicio 3).

```
Profesor profesor = new Profesor("30.000.001", "profesor", 100, "+54 (0375) 123456789");
```

```
Curso c1 = new Curso("Programación Java", profesor, 2, "virtual");  
Curso c2 = new Curso("Electricidad", profesor, 3, "presencial");  
Curso c3 = new Curso("Carpintería", profesor, 1, "presencial");
```

```
Alumno a1 = new Alumno("12.345.671", "Juan", "Posadas", 2016);  
Alumno a2 = new Alumno("12.345.672", "Mirta", "El Dorado", 2017);  
Alumno a3 = new Alumno("12.345.673", "María", "Posadas", 2018);
```

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

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

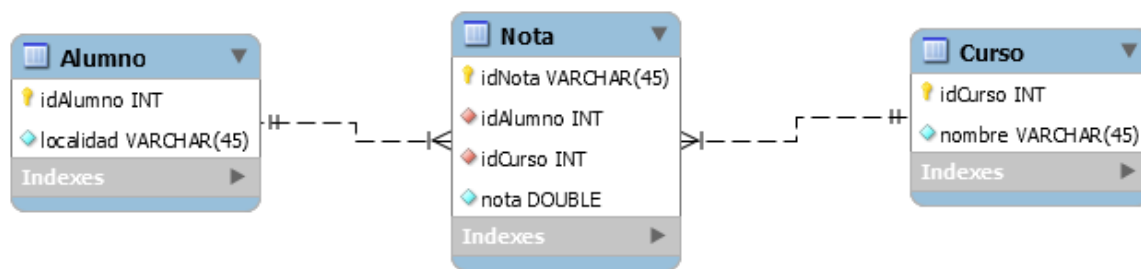
```
c3.agregarAlumno(a1);  
c3.agregarAlumno(a2);  
c3.agregarAlumno(a3);
```

```
System.out.println(c1.getAlumnos().size());  
System.out.println(c2.getAlumnos().size());  
System.out.println(c3.getAlumnos().size());
```

12. Qué imprime?

Ejercicio 5. Consulta SQL

Se ha modelado el siguiente diagrama de Entidad Relación.



Consulta SQL

El departamento de estadísticas de la UNAM desea realizar un relevamiento del desempeño del alumnado en el curso de Java en el interior de la provincia. Dado el diagrama de entidades y relaciones parcial del sistema, escribir una consulta que liste para cada localidad el promedio de notas de los alumnos de dicha localidad en el curso con nombre "Java". Como se trata del interior de la provincia, ignorar la localidad "Posadas". Además, como se desea dar un premio a las mejores localidades, solo considerar los promedios de nota mayor o igual a 7. Indique el resultado de aplicar su consulta a los siguientes datos.

Alumno	
idAlumno	localidad
1	Posadas
2	Oberá
3	Oberá

Nota			
idNota	idAlumno	idCurso	nota
1	1	1	8
2	2	1	10
3	3	1	4

Curso	
idCurso	nombre
1	Java
2	Carpintería
3	Construcción en Seco

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 "localidad".

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**