

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 indicalo aquí. El mismo te llegará en forma digital, al correo electrónico que indicaste al comienzo de este formulario
Marca solo un óvalo.

- ☐ Sí
- ☐ No

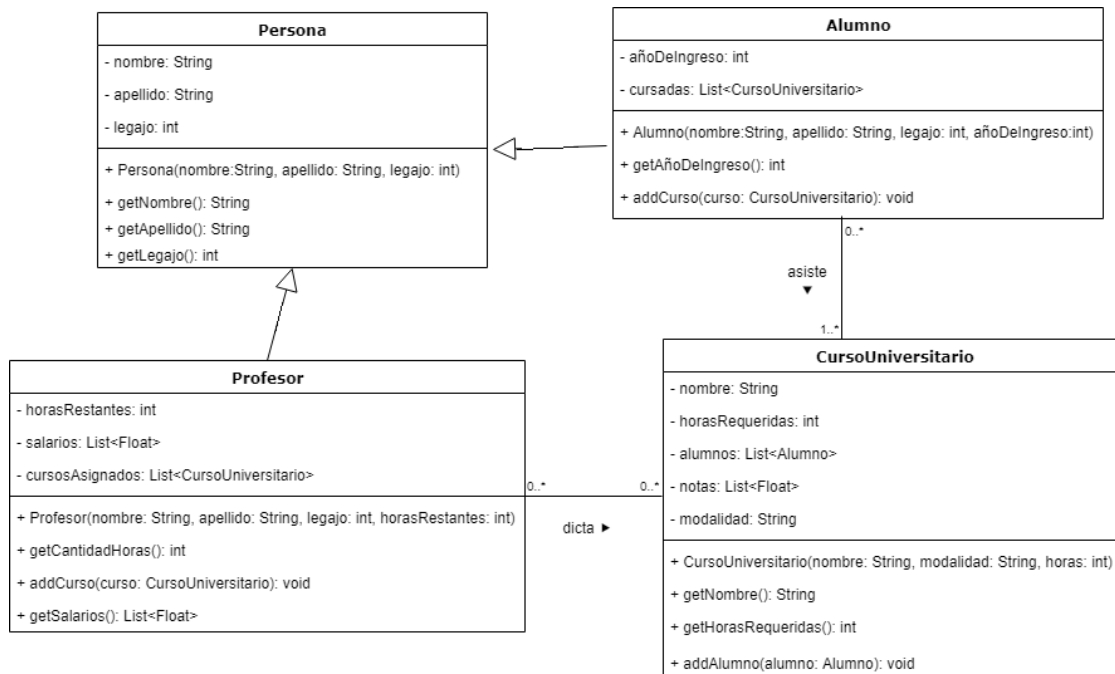
Contexto del examen - Cursos Universitarios

La Universidad de la Mesopotamia (UNM) abrió recientemente un programa de formación continua para la comunidad universitaria. Desde la UNM surgió la necesidad de desarrollar un sistema para administrar los cursos universitarios que se dictan en cada cuatrimestre. 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

Profesores del programa de la UNM realizaron un diagrama UML preliminar del sistema y nos pidieron que implementemos la clase CursoUniversitario según el diagrama (el resto de las clases serán implementadas por la UNM). Los profesores nos indicaron que, una vez que hayamos implementado la clase CursoUniversitario, ejecutarán el siguiente código para comprobar que no existen errores:

```
CursoUniversitario curso=new CursoUniversitario ("Inteligencia Artificial", "Presencial", 40);
Alumno alumno = new Alumno("Juan", "Pérez", 20546, 2019);
curso.addAlumno(alumno);
```



9. Implemente la clase CursoUniversitario

Ejercicio 2. Implementar un método a partir de un enunciado

Programar en Java la funcionalidad para obtener el promedio de horas de todos los cursos universitarios de una determinada modalidad (ej. virtual, presencial, etc.) realizados por un determinado alumno. Puede suponer que el alumno realizó al menos un curso universitario de la modalidad buscada.

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

Desde la administración de la UNM decidieron limitar la cantidad de cursos universitarios que cada profesor puede dictar de modo de asegurar la calidad en el dictado de sus cursos. Para ello, nos solicitaron que modifiquemos el método llamado `addCurso` en la clase `Profesor`. El método solo debe agregar el curso universitario a la lista `cursosAsignados` si el profesor aún cuenta con horas disponibles por semana. Tenga en cuenta que cada curso universitario contiene las horas semanales requeridas. En caso de cumplirse esta condición, el método deberá agregar el curso universitario en la lista `cursosAsignados`, decrementar las horas disponibles del profesor 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 profesores de la UNM sobrescribieron el método `toString` en las clases `Persona` y `Profesor` de la siguiente manera:

Clase `Persona`

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

Clase `Profesor`

```
@Override
public String toString() {
```

```

    return super.toString()+" "+ this.horasRestantes+" "+ this.cursosAsignados.size();
}

```

Considerando estos cambios, ¿Qué imprimirá el programa al ejecutar el siguiente código? Recuerde que el método addCurso asigna un curso al Profesor siempre y cuando no se haya alcanzado el número máximo horas disponibles para ese profesor y decrementa las horas disponibles del profesor.

```

CursoUniversitario c1=new CursoUniversitario ("Programación", "Presencial", 4);
CursoUniversitario c2=new CursoUniversitario ("Bases de Datos", "Virtual", 2);
CursoUniversitario c3=new CursoUniversitario ("Técnicas", "Presencial", 6);

```

```

Profesor p1=new Profesor ("Daenerys", "Targaryen", 123456, 40);
Profesor p2=new Profesor ("Jon", "Snow", 123457, 30);
Profesor p3=new Profesor ("Cersei", "Lannister", 123458, 20);

```

```

p1.addCurso(c1);
p2.addCurso (c2);
p3.addCurso (c3);

```

```

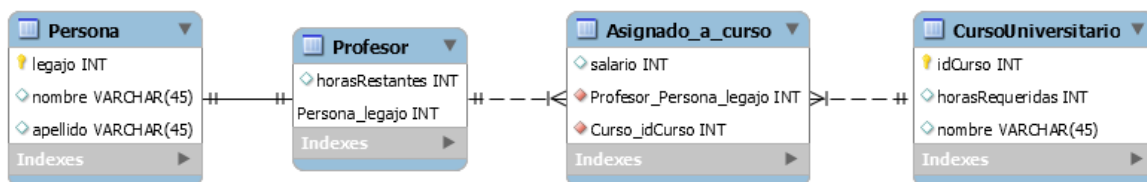
System.out.println(p1);
System.out.println(p2);
System.out.println(p3);

```

12. Qué imprime?

Ejercicio 5. Consulta SQL

Se ha modelado el siguiente diagrama de Entidad Relación.



Consulta SQL

Dado el diagrama de entidad-relación parcial, escriba una consulta SQL que liste para cada profesor el promedio de los salarios de los cursos universitarios que dicta. Solo se deben tener en cuenta aquellos cursos universitarios que requieren 20 horas o menos. Los resultados deben estar ordenados ascendentemente por el legajo de los profesores. Además, dadas las siguientes tuplas de ejemplo, determinar el resultado de la consulta.

Persona		
123456	Daenerys	Targaryen
123457	Jon	Snow
123458	Cersei	Lannister
Profesor		
50	123456	
65	123457	
75	123458	

Asignado_a_curso		
12.000	123456	1
10.000	123457	1
16.000	123458	1
9.000	123456	2
12.000	123457	2
12.000	123458	2
10.000	123456	3
8.000	123457	3
2.000	123458	3
CursoUniversitario		
1	10	Programacion
2	25	Bases de datos
3	15	Tecnicas

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 "horasRequeridas".

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