

INFORMACIÓN PARA EL ALUMNO ANTES DE LA APLICACIÓN DEL EXAMEN DE CERTIFICACIÓN

1. Información del instrumento

Plan de Estudio	Desarrollo de Aplicaciones Full Stack Java Trainee
-----------------	--

2. Características de la aplicación

Tiempo Total	360 minutos
Tipo de Trabajo	Individual
Lugar	Sesión sincrónica

3. Descripción del examen

El examen consiste en la resolución de un caso de negocio, en donde usted deberá construir una aplicación web dinámica en lenguaje Java que satisfaga los requerimientos entregados. Dichos requerimientos estarán alineados con las siguientes competencias:

- Codificar piezas de software de baja/mediana complejidad en Java utilizando el paradigma de orientación a objetos para resolver una problemática de acuerdo a las buenas prácticas de la industria
- Construir consultas de obtención, manipulación y definición de datos en una base de datos relacional utilizando lenguaje SQL para satisfacer un problema planteado
- Construir una página web responsiva básica utilizando HTML, CSS y JavaScript acorde a las buenas prácticas de la industria
- Construir aplicaciones web dinámicas utilizando el entorno de desarrollo Java que resuelven problemáticas de la industria a partir de especificaciones provistas y utilizando buenas prácticas de la industria
- Construir aplicaciones empresariales, que resuelven problemáticas de la industria, utilizando el entorno JEE a partir de especificaciones provistas y acorde a las buenas prácticas de la industria

El examen es de ejecución individual, es decir, debe ser resuelto solamente por usted, no puede intercambiar información con sus compañeros ni con personas externas. Sin embargo, usted puede consultar todos sus apuntes y bibliografía con que cuente el día de la ejecución, asimismo, podrá consultar internet libremente si lo requiere.

Se le recomienda verificar antes de iniciar el examen que usted cuenta con las herramientas requeridas para la resolución del caso, esto es:

- Entorno Integrado de Desarrollo
- Versiones de Java adecuadas
- Base de datos disponible para su conexión

Al momento de la ejecución del examen, usted recibirá el caso, que está compuesto por los siguientes componentes:

- Descripción del caso
- Modelo de Datos
- Script de creación y poblamiento del modelo de datos

Al finalizar su trabajo, usted deberá comprimir el código fuente de su proyecto y seguir las instrucciones para subirlo a la plataforma de evaluación.

4. Componentes de la evaluación

Evidencias	La(s) evidencia(s) de la evaluación será(n): <ul style="list-style-type: none"> • De Producto, que consiste en: <ul style="list-style-type: none"> ○ Código fuente ○ Aplicativo en funcionamiento
Situación Evaluativa	La(s) situación(es) en la que se evaluará el desempeño consistirá(n) en: <ul style="list-style-type: none"> • Ejecución Práctica
Instrumento	Para recoger la información se utilizará: <ul style="list-style-type: none"> • Rúbrica

5. Puntajes y Nota

Puntaje Total	80	Nota	7.0
Puntaje Mínimo	48		4.0

Presentación del caso.

La Prestigiosa University Of The Life lo ha contratado para modernizar su módulo académico online, el cual es responsable de matricular a los alumnos, registrar a los profesores, crear los cursos, inscribir a los alumnos en cada curso, creación y registro de evaluaciones y de asistencia que se realizan de manera Online la cual ha tenido un crecimiento importantes en los últimos años y es necesario realizar mejoras a lo que poseen, los alumnos se inscriben a una especialidad y toman los cursos que posea la especialidad , los profesores dictan los ramos via web y con videos asincrónicos, luego los alumnos rinde las prueba que tiene cada ramo y estos deben aprobar con una nota mínima de 4 la escala es del 1 al 7 :

- El sistema tendrá los siguientes roles
 - Administrativo
 - Profesor.
 - Alumno.

Rol	Matricular	Inscripción Alumnos	Crear Cursos	Registra Nota	Registra Asistencia	Leer Notas
Administrativo	SI	SI	SI	SI	SI	SI
Profesor	NO	NO	NO	SI	SI	SI
Alumno	NO	NO	NO	NO	NO	SI

Se evaluará el manejo de:

- Deberá ser realizado usando Spring Framework.
- Utilizar el patrón de diseño MVC.
- Bootstrap 5 para modelar las vistas usando JSP y/o Thymeleaf. Emplear la utilización de JSP y Taglib para modelar las vistas solicitadas.
- Consultas SQL a las tablas mediante JdbcTemplate o Spring Data JPA (u otro).
- La autenticación deberá ser validada con Spring Security en el proyecto.
- Crear un servicio REST que permita acceder al stock de productos por sucursal.

Entregables del proyecto:

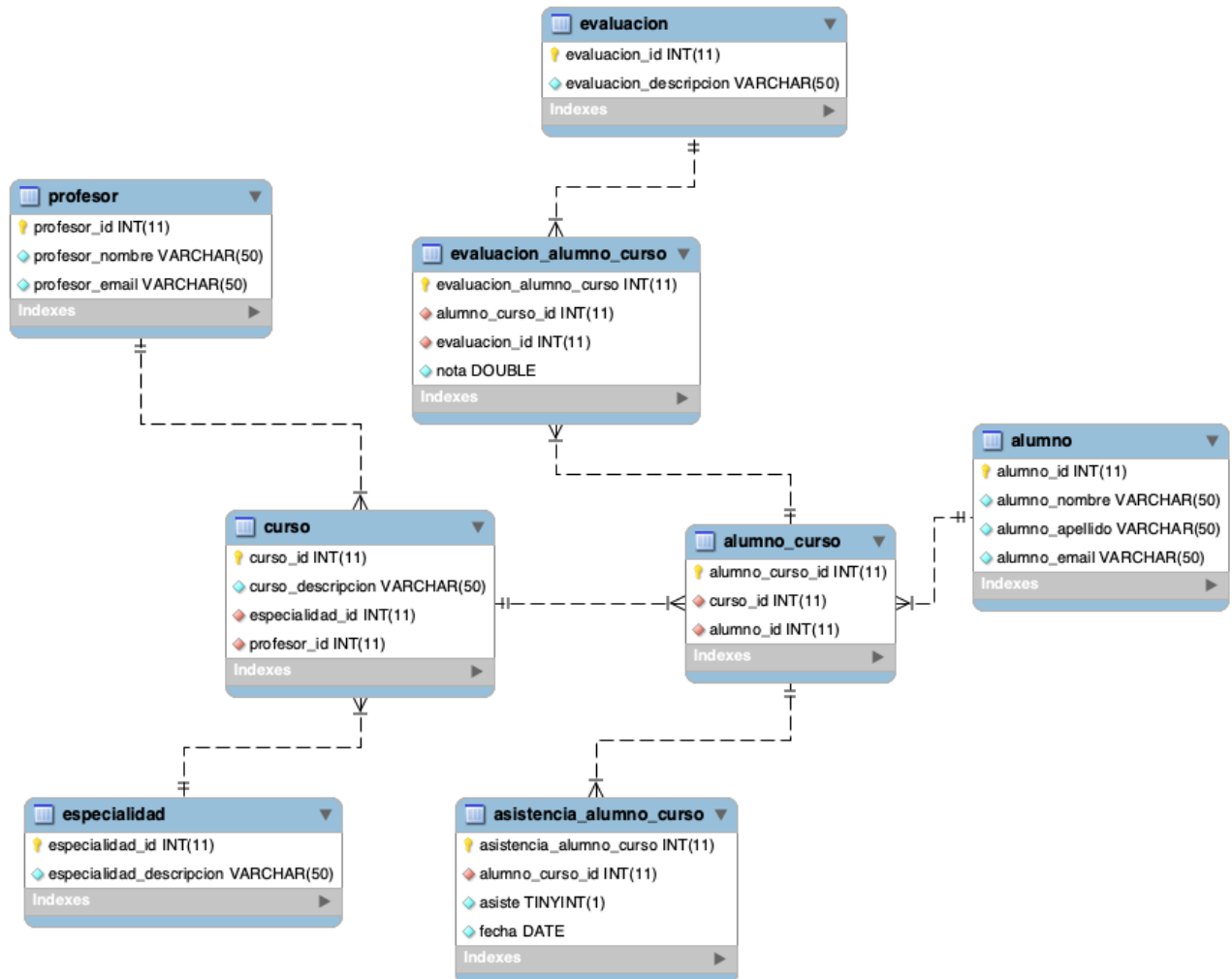
- Fase 1: Resolución de las consultas según un Modelo de Datos Relacional y script SQL para ejecución.
- Fase 2: Aplicación de escritorio en Java
- Fase 3: Implementar vistas y una API REST según las consignas a continuación.

Otras consideraciones:

- Debe tener en consideración cumplir con las buenas prácticas en programación.
- Deberá ser **entregado el código fuente a GitHub** de forma privada para revisión.

Fase 1: Consultas SQL

Para la resolución de este ítem se requiere conocer el siguiente modelo relacional (se incluye un script para la generación de las tablas y datos):



De éste modelo relacional se requerirán realizar las siguientes consultas SQL para generar reportes de interés:

1. Listar Los Cursos disponible por especialidad , filtrado por id de especialidad.

curso_descripcion	especialidad_descripcion
Sql de 0 a Experto	Informática
Javascript de 0 a Experto	Informática
Python de 0 a Experto	Informática

2. Listar el nombre de cada curso y la cantidad de alumnos inscritos.

curso_descripcion	alumnos
Contabilidad I	21
Contabilidad II	18
Contabilidad III	33
Diseño 3D	15
Diseño I	12
Introducción al Diseño	17
Javascript de 0 a Experto	12
Python de 0 a Experto	14
Sql de 0 a Experto	16

3. Listar todos los alumnos indicando nombre, apellido, email, cantidad de ramos tomados para una especialidad (en el ejemplo se tomó el id de especialidad 3).

alumno_nombre	alumno_apellido	ramos
Brent	Woolliams	1
Millicent	Ackers	1
Floris	Dmtrovic	1
Marcile	Dudden	1
Boothe	MacKissack	1
Salomi	Scutcheon	1
Sharai	Dominighi	1
Brantley	Yorath	2
Hana	Ramsay	2
Rick	Anstey	2
Cindee	Newiss	2
Catie	Antham	2
Arlette	Cribbins	2
Bar	Wardall	2
Mamie	Bengtzen	2
Ivette	Keats	2
Daisy	Capun	2
Noemi	Dunston	2
Wilma	Ville	2
Wilek	Credland	2
Piggy	Clifforth	2
Livvy	Sprionghall	2
Amandie	Wynrahame	2
Konstantine	Cristofaro	2
Wayne	Randlesome	3

4. Listar la cantidad de inasistencias de todos los curso para una fecha, (en el ejemplo se tomó el 2024-05-01).

curso_descripcion	count(*)
Contabilidad III	3
Python de 0 a Experto	1
Sql de 0 a Experto	1

5. Agrupar la cantidad de asistencias que se han realizado un alumno para un curso determinado, mostrando cantidad, el el nombre del alumnos y el curso, estableciendo el orden de forma descendiente por la cantidad de asistencias

alumno_nombre	alumno_apellido	curso_descripcion	asistencias
Brent	Woolliams	Contabilidad III	7
Rois	Slocum	Contabilidad III	7
Juanita	Clapshaw	Contabilidad I	7
Catie	Antham	Contabilidad III	6
Geoffrey	Binham	Contabilidad II	6
Anastassia	Vockins	Javascript de 0 a Experto	6
Konstantine	Cristofaro	Contabilidad III	6
Maryl	Garraway	Sql de 0 a Experto	6
Rick	Anstey	Diseño I	6
Alena	Eykel	Contabilidad III	6
Livvy	Sprionghall	Contabilidad III	6
Agace	Veart	Python de 0 a Experto	6
Corrina	Gogay	Contabilidad I	6
Wayne	Randlesome	Introducción al Diseño	6
Maryl	Garraway	Python de 0 a Experto	5
Briana	Rubinov	Contabilidad III	5
Mamie	Bengtsen	Introducción al Diseño	5

Fase 2: Aplicación escritorio

Los algoritmos a implementar serán los siguientes:

- Implementar un algoritmo que permita saber que cursos aprueba un alumno, para ello debe calcular el promedio de sus cursos y si la nota el promedio mayor a 4 lo aprueba, debe entregarle el id de alumno.

Ej

Curso 1	Curso 2	Curso 3	Promedio	Estado Final
5.5	3.8	4.5	4,6	Aprueba

- Implementar un algoritmo que permite saber que cursos deben estar con mas de 25 alumnos y se deba crear una nueva sección , para ello debe indicar el id de especialidad

Se deberán implementar dos interfaces: una para la validación de notas y la otra validación del cupos por curso

Ejemplo de ingreso de valores por la consola

```
Indique el id del alumno : 25
La notas de Jose Toro son :
Curso 1 : 5.5
Curso 2 : 3.8
Curso 3 : 4.5
Promedio : 4.6
Estado Final : Aprueba
```

```
<terminated> Main (8) [Java Application] /Users/juanpabloduran/Library/Java/JavaVirtualMachines/openjdk-21.0.2/Contents/Home/bin/java (26-05-2024 21:50:28 - 21:50:31) [pid: 72605]  
Indique el id de la especialidad : 2  
Contabilidad III Tiene 33 alumnos debe habilitarse una nueva seccion
```

Una vez que se ha implementado todo lo anterior, realizar pruebas al código para determinar el correcto funcionamiento de los algoritmos.

Fase 3: Vistas e implementación de Servicio Rest

Crear una vista que filtre por Especialidad y Curso para saber los todos los datos de los Alumnos y su promedio

Prototipo de Diseño:

Especialidad

Contabilidad

Curso

Contabilidad I

Buscar

#	Nombre	Apellidos	Email	Curso	Promedio
1	Juan	Perez	jperez@email.cl	Contabilidad I	5.8
2	Maria	Rojas	mrojas@email.cl	Contabilidad I	6.4

Se pide:

1. Desplegar el listado de las Especialidades en el primer combobox, ordenado alfabéticamente, con valores que provengan de la base de datos
2. Desplegar el listado de tipo Cursos en el segundo combobox, ordenado alfabéticamente con valores que provengan de la base de datos
3. Desplegar el listado de los registros en base a lo solicitado usando los filtros de Especialidad y Curso.
4. Requerimientos ya mencionados en el documento.

Disponibilice un servicio REST que permita obtener la misma información para la vista antes mencionada.