

Modelado JPA/Hibernate

Contenido

Especificaciones.....	2
Requisitos	3
Diseño.....	4
Requisitos generales.....	4
Requisitos específicos.....	5
Navegabilidad	5
Consultas	6
ANEXO I: Código de ejemplo en una clase componente.....	7
ANEXO II: Tests unitarios de ejemplo	8
ANEXO III: Algunos datos de prueba	8
Departamentos	8
Grupos de desarrollo	9
Cursos de formación.....	9
Proyectos.....	9
Empleados.....	10
Usuario	12
Becarios.....	13

Especificaciones de negocio

En una empresa trabajan empleados, de los que se quiere conocer su nombre, apellidos, fecha de nacimiento, género, ~~email, teléfono móvil~~, dirección postal, período en que ha trabajado en la empresa, es decir, la fecha de contratación y la fecha de cese, en el caso de que ya hubiera abandonado la empresa. El cese puede ser por diferentes motivos (abandono de la empresa, por despido, por jubilación, incapacidad médica permanente) se debe también consignar.

También se necesita almacenar el salario y comisión actual, así como la lista de puestos desempeñados en la compañía, con sus períodos y salarios (se asume que durante el periodo de desempeño de un puesto el salario no varía).

La organización de la empresa es jerárquica, es decir, cada empleado tendrá un único jefe directo (a excepción del presidente de la empresa, que no tiene un superior), y un empleado puede tener subordinados.

De cada empleado se almacena, igualmente, el banco y la cuenta bancaria en la que se realizará el pago de la nómina. Al insertar la cuenta se validará que es correcta. Adicionalmente, un empleado puede tener asociada una tarjeta de crédito a cuenta de la empresa con la que hará los pagos asociados su actividad laboral. De cada tarjeta de crédito se almacenará el número, el CVV/CVC y la fecha de caducidad.

Cada empleado cobra una nómina mensual. De cada nómina se registra el mes y año al que corresponde. Cada nómina está compuesta de líneas de nómina (al menos una) que son los conceptos por los que al empleado se le ingresan o le retienen cantidades. Los ingresos se registrarán como cantidades positivas; las retenciones, como cantidades negativas. El líquido de una nómina es el resultante de sumar/restar todas las cantidades de las diferentes líneas de nómina.

Los empleados pueden tener una cuenta para acceder a la aplicación informática corporativa. La información de esta cuenta que se desea almacenar es el nombre de usuario, la contraseña y la fecha del último acceso. A efectos de validación, además de la contraseña, se almacenará la confirmación de la misma, pero este será un atributo transitorio que no se persistirá en la base de datos. Dos empleados no pueden tener un mismo nombre de cuenta (*login*), pero sí podría coincidir que tuvieran la misma contraseña.

Además, en la empresa trabajan becarios, de los que se quiere conocer su nombre, apellidos, fecha de nacimiento, género, ~~email, período de servicio en la empresa~~. Además, su dirección personal y el nombre y la dirección del centro de estudios al que pertenecen. Cada becario tiene asignado un único empleado que es su mentor. Un empleado, a su vez, puede mentorizar a un becario.

Los empleados realizan cursos de formación para actualizar sus conocimientos en diferentes ámbitos. A su vez, estas formaciones las imparten empleados que están cualificados para ello. Así, son empleados los que imparten estos cursos de formación

(ponentes) a otros empleados (estudiantes). De cada curso se quiere registrar su nombre, las horas que dura y la lista de empleados que están cualificados para impartirlos. Para cada curso hay una serie de ediciones, de las que se quiere saber su período de desarrollo y su ponente. También se quiere registrar qué empleados asisten a cada edición de un curso de formación, así como la calificación que obtienen en esa edición. Un empleado no puede asistir como alumno a una edición del mismo curso de formación del que es ponente. Además, cada empleado no podrá cursar simultáneamente más de un curso de formación. Tampoco podrá impartir más de un curso de formación simultáneamente.

Cada empleado trabaja en un único departamento, aunque temporalmente podría no estar asociado a ninguno. Cada departamento tiene un nombre, un código único, una localidad donde se ubica y un presupuesto anual. Además, cada departamento puede tener a un empleado como jefe. A su vez, un empleado no podrá ser simultáneamente jefe de más de un departamento.

Un empleado puede colaborar en múltiples proyectos, desempeñando el mismo o distinto rol. Cada proyecto tiene asociado un nombre único ~~y un período de desarrollo~~. Se quiere conocer en qué proyectos ha trabajado cada empleado, los períodos en los que lo hizo y el rol que desempeñó en cada período. Un empleado puede trabajar en el mismo proyecto en dos periodos diferentes, desempeñando el mismo o distinto rol.

En la presa existen grupos de desarrollo personal (ej. trabajo en equipo, meditación, expresión corporal...), a los que los empleados pueden pertenecer. Un empleado puede pertenecer a varios de estos grupos, y en un mismo grupo puede haber varios empleados apuntados.

Requisitos

- Como punto de partido, se creará un modelo de datos con MySQL Workbench.
- Se deben crear un conjunto de clases anotadas (POJOs) que representen las especificaciones indicadas a continuación. Las clases pueden ser entidades o no.
- Para cada entidad, se creará una interfaz repositorio en la que se describirán los métodos necesarios para realizar las operaciones requeridas sobre la base de datos.

Ej. Para la entidad Departamento se creará *DepartamentoRepository*:

```
@Repository
public interface DepartamentoRepository extends
JpaRepository<Departamento, UUID> {}
```

- No se codificará ninguna clase controlador. Para hacer las pruebas de funcionamiento, se añadirá código en clases componente (véase [ANEXO I](#)) y se construirán test unitarios (véase [ANEXO II](#)).
- Se aportan algunos datos de prueba (véase [ANEXO III](#)).

Diseño

Requisitos generales

1. Base de datos:

Los nombres de tablas y campos de la base de datos:

- Estarán en minúscula y singular (**ej.** departamento).
- Si se trata de un nombre compuesto, se separarán las palabras por el carácter subrayado (**ej.** fecha_inicio).

La clave primaria de toda tabla maestra será un campo llamado id de tipo UUID.

Los nombres de las restricciones existentes sobre campos de tablas:

- Las claves primarias se nombrarán de la forma PK_<tabla> (**ej.** PK_departamento).
- Las claves ajenas se nombrarán de la forma FK_<tabla_origen>_<tabla_destino>_<campo_en_tabla_origen> (**ej.** FK_empleado_departamento_id_departamento).
- Las restricciones de unicidad se nombrarán de la forma UQ_<nombre_tabla>_<nombre_campo> (**ej.** UQ_depatamento_nombre)
- Las retenciones de chequeo se nombrarán de la forma CK_<tabla>_<campo>_<condición_chequeo> (**ej.** CK_empleado_oficio_nonulo)

Los índices se nombrarán de la forma IDX_<tabla>_<campo> (**ej.** IDX_empleado_id_departamento).

En tablas de tamaño potencialmente grande los campos que sean clave ajena se indexarán para agilizar las búsquedas.

Ej. En la tabla movimientos_bancarios se indexará el campo id_cuenta que referencia la cuenta corriente a la que pertenece el movimiento.

2. Entidades

Cada entidad de negocio tendrá un identificador del tipo UUID. El nombre del atributo será siempre **id**, llamándose de igual manera el campo asociado en la base de datos (véase comentario en la sección de bases de datos).

Además del identificador, toda entidad tendrá un atributo sobre el que se establecerá una restricción de unicidad, y permitirá identificar de manera única a cada instancia de la entidad, además del ya citado atributo identificador.

Cuando varias entidades contengan una serie de atributos comunes, se implementarán como una jerarquía de clases mediante herencia.

3. Otras clases

Si es necesario que la clase tenga un atributo identificador, éste será de tipo Long.

Se utilizarán clases auxiliares que contengan información susceptible de ser empleada en diversas entidades, como períodos de tiempo, direcciones postales, tarjetas de crédito, cuentas bancarias...

Requisitos específicos

1. Empleado

La información económica de cada empleado se almacenará en una tabla diferente de la tabla principal de empleados. Esto es, salario, comisión, cuenta corriente, tarjeta de crédito, nóminas...

El género de los empleados, becarios... será de tipo enumerado.

Se quiere dar la posibilidad de eliminar lógicamente los empleados, de manera que la información se mantenga físicamente en la base de datos pero no se visualice al hacer consultas. Para ello, se dispondrá de un campo con la fecha de eliminación que, de ser nulo, implicará que el empleado no está borrado lógicamente, y de tener una fecha, indicará que el empleado se borró lógicamente en esa fecha.

Si un empleado ha sido borrado lógicamente, no debe aparecer en el listado de empleados de un departamento, o en el listado de empleados de un proyecto, o en el listado de los empleados que participan en un grupo de desarrollo.

2. Usuario

La información relativa al usuario de la aplicación informática se considerará una entidad de negocio independiente del Empleado.

3. TarjetaCredito

En particular, con la clase TarjetaCredito se desarrollarán las validaciones necesarias para que no se puede dar de alta una tarjeta de crédito si:

- la fecha de expiración es en el pasado.
- no cumple el algoritmo de Luhn (véase Validate Credit Card Numbers)

Navegabilidad

Se va a utilizar la expresión “conoce” para indicar que existe navegabilidad desde una instancia de una entidad hacia la instancia (o instancias) de otra entidad.

- Cada empleado “conoce” directamente quien es su empleado jefe, pero un empleado no conoce directamente quiénes son sus empleados subordinados.

- Cada empleado conoce la lista de puestos que ha desempeñado, pero no al revés.
- Cada empleado “conoce” directamente su departamento, y también el departamento “conoce” a sus empleados.
- Cada becario conoce a su mentor, pero no al revés.
- Cada empleado conoce a su usuario de la aplicación, pero no el revés.
- Cada empleado conoce su lista de nóminas, pero no al revés.
- Cada nómina “conoce” su lista de líneas de nómina, pero no al revés.
- Cada curso de formación “conoce” la lista de empleados que están cualificados para impartirlo, pero no al revés.
- Cada curso de formación “conoce” su lista de ediciones, y también al revés.
- Cada edición de un curso de formación “conoce” a su empleado ponente, pero no al revés.
- Cada empleado “conoce” la lista de proyectos en los que ha participado, y también al revés.
- Cada grupo de desarrollo “conoce” la lista de empleados que participan en el, pero no al revés.

Consultas

Se insertarán una serie de datos de prueba para realizar las consultas, es decir, algún departamento, algún empleado, algún proyecto...

1. Listado de empleados de un departamento.
2. Lista de empleados que hayan trabajado en la empresa en un periodo de tiempo recibido por parámetro, es decir, que el período pasado por parámetro coincida, al menos en un día, con algún periodo en el que el empleado haya trabajado en la empresa.
3. Listado de empleados cuyo salario actual se encuentre en un rango de valores.
4. Listado de empleados subordinados de uno dado, recibiendo como parámetro el identificador de empleado del jefe.
5. Sueldo líquido de un empleado en un mes determinado, recibiendo como parámetros el identificador del empleado, así como el año y el mes de la nómina.
6. Sueldo bruto anual de un empleado, recibiendo como parámetros el identificador del empleado y el año.
7. Datos económicos de un empleado (cuenta corriente y tarjeta bancaria), recibiendo como parámetro el identificador del empleado.
8. Listado de apellidos de empleados, su cuenta asociada (en caso de tenerla) y la fecha de último acceso.
9. Datos del becario mentorizado por un empleado, dado como parámetro el identificador del empleado.

10. Listado de cursos de formación para los que está cualificado a impartir un empleado, recibiendo como parámetro el identificador del empleado.
11. Listado de empleados cualificados para impartir un curso de formación, recibiendo como parámetro el identificador del curso.
12. Listado de las ediciones de cursos de formación a las que ha asistido un empleado como alumno, recibiendo como parámetro el identificador del empleado.
13. Listado de ediciones de un curso de formación, recibiendo como parámetro el identificador del curso.
14. Número de horas de formación que ha realizado un empleado en un periodo de tiempo, recibiendo como parámetros el identificador del empleado y el periodo de tiempo.
15. Listado de empleados que han trabajado en un proyecto, con sus fechas de participación y roles. Se recibirá como parámetro el identificador del proyecto.
16. Listado de grupos de desarrollo a los que pertenece un empleado. Se recibirá como parámetro el identificador del empleado.

ANEXO I: Código de ejemplo en una clase componente

```
@Component
public class InsercionManipulacionDatos {

    @Autowired
    DepartamentoRepository departamentoRepository;

    @Bean
    public static void eliminaDepartamentos() {
        departamentoRepository.deleteAll();
    }

    @Bean
    private static void pueblaDepartamentos() {
        // datos de departamentos
        Departamento d1 = new Departamento();
        d1.setCodigo("10"); d1.setNombre("Ventas"); d1.setLocalidad("Huelva");
        d1.setPresupuesto(BigDecimal.valueOf(162000));
        departamentoRepository.save(d1);

        Departamento d2 = new Departamento();
        d2.setCodigo("20"); d2.setNombre("Almacén"); d1.setLocalidad("Soria");
        d2.setPresupuesto(BigDecimal.valueOf(87000));
        departamentoRepository.save(d2);
    }

    @Bean
    private static void listaDepartamentos() {
        System.err.println("Listado de departamentos");
        departamentoRepository.findAll() // se asume implementado toString()
            .forEach(dept -> System.err.println(dept));
    }
}
```

ANEXO II: Tests unitarios de ejemplo

17. **Ej 1.** Operaciones sobre las entidades Banco y CuentaCorriente, relacionadas **bidireccionalmente** por una relación **@OneToMany** (se puede consultar el ejemplo completo en la referencia [Ejemplo de relación OneToMany en Hibernate](#), aunque puede estar modificado por facilitar la comprensión del ejemplo):

```
@SpringBootTest
public class CuentaCorrienteRepositoryTest {
    @Autowired
    private CuentaCorrienteRepository cuentaCorrienteRepository;
    @Autowired
    private BancoRepository bancoRepository;
    @Test
    // public void
    // when_save_new_bank_with_multiples_accounts_then_bank_is_saved_correctly() {
    public void
    al_grabar_nuevo_banco_con_varias_cuentas_se_graban_todos_objetos_correctamente() {
        Banco banco = new Banco("Banco del mediodía");
        CuentaCorriente cc1 = new CuentaCorriente(BigDecimal.valueOf(450));
        banco.addCuentaCorriente(cc1);
        CuentaCorriente cc2 = new CuentaCorriente(BigDecimal.valueOf(325));
        banco.addCuentaCorriente(cc2);
        bancoRepository.save(banco);
    }
}
```

ANEXO III: Algunos datos de prueba

Se aportan una serie de datos de prueba con el único fin de poder hacerse una idea de la información que se almacena de cada entidad. Los datos pueden estar incompletos. No es necesario usar exactamente estos datos para hacer las pruebas.

Departamentos

Nombre: Ventas Código: 10 Presupuesto: 150000 Jefe: Pedro Medario
Nombre: Almacén Código: 20 Presupuesto: 60000 Jefe:
Nombre: Tecnología Código: 20

Presupuesto: 33000

Jefe: Marta Padera

Grupos de desarrollo

Nombre: Meditación

Nombre: Trabajo en equipo

Nombre: Expresión corporal

Nombre: Liderazgo

Cursos de formación

Nombre: SpringBoot, un Frameworks de microservicios

Duración: 40 horas

Empleados cualificados: Ana Lucía Tica,

Ediciones:

- 01/01/2000 a 31/01/2000

· Impartido por Ana Lucía Tica

· Asisten:

· José Ramón Dongo, obteniendo una calificación de 3.

· Adela Drador, obteniendo una calificación de 7.

- 01/01/2000 a 31/01/2000

· Impartido por Ana Lucía Tica

· Asisten:

· José Ramón Dongo, obteniendo una calificación de 3.

· Adela Drador, obteniendo una calificación de 7.

Nombre: Angular, un Frameworks del lado cliente

Duración: 60 horas

Ediciones:

- 01/03/2001 a 30/05/2001

· Imparte: Federico Najero

· Asisten:

· José Ángel Atila, obteniendo una calificación de 7.

· Lurdes Enfreno, obteniendo una calificación de 8.

Proyectos

Nombre: Mariposa

Período:

- Inicio: 01/01/2000

- Fin: 15/02/2003

Empleados que trabajan:

* Juan Pedro Medario:

- de 01/01/2000 a 30/06/2000, como analista junior

- de 01/09/2000 a 30/11/2000, como analista senior

* José Ramón Dongo

- de 01/01/2000 a 30/06/2001, como gerente
- de 21/08/2002 a 15/02/2003, como gerente
- * Ana Lucía Tica
- de 01/07/2001 a 20/08/2002, como gerente
- * Adela Drador
- de 01/01/2000 a 15/02/2003, como desarrollador senior

Empleados

Nombre: Juan Pedro
Apellidos: Medario
Fecha de nacimiento: 01/01/1981
Género: Masculino
Dirección:

- Tipo vía: Calle
- Vía: Percebe
- Número: 13
- Piso: 5
- Puerta: A
- Localidad: Guadalajara
- CP: 19123
- Región: Castilla La Mancha
- País: España

Período de contratación:

- Fecha contrato: 01/01/2010
- Fecha cese: 31/12/2020
- Motivo: Abandono de la empresa

Puestos:

- Puesto: Jefe de proyecto
 - Periodo: 01/01/2010 a 31/12/2012
 - Salario: 2000
- Puesto: Gerente
 - Periodo: 01/01/2013 a 31/12/2015
 - Salario: 2500
- Puesto: Director del departamento de tecnología
 - Periodo: 01/01/2015 a 31/12/2018
 - Salario: 4000
- Puesto: Presidente
 - Periodo: 01/01/2019 a 31/12/2020
 - Salario: 6000

Comisión: 600
Jefe:
Grupos de desarrollo:

- Liderazgo
- Meditación

Cuenta corriente:

- Cuenta bancaria: ES12.3456.7890.1234.5678.9012

<ul style="list-style-type: none"> - Banco: Banco Nocado <p>Tarjeta de crédito:</p> <ul style="list-style-type: none"> - Número: 0987.6543.2109.8765 - Fecha de caducidad: 01/01/2021 - CVC: 123 <p>Nóminas:</p> <ul style="list-style-type: none"> - Fecha: 01/2010 - Líneas: <ul style="list-style-type: none"> · Sueldo: +2500 · Retenciones pensión: -300 · Retenciones servicio sanitario: -50 · Impuestos: -150 - Ingreso líquido: 2000
<p>Nombre: Marta</p> <p>Apellidos: Padera</p> <p>Fecha de nacimiento: 02/02/1982</p> <p>Género: Femenino</p> <p>Dirección:</p> <ul style="list-style-type: none"> - Tipo vía: Avenida - Vía: Aguilucho - Número: 4 - Piso: 2 - Puerta: D - Localidad: Madrid - CP: 28456 - Región: Madrid - País: España <p>Período:</p> <ul style="list-style-type: none"> - Fecha contrato: 15/06/2015 - Fecha cese: - Motivo: <p>Jefe:</p> <p>Grupos de desarrollo:</p> <ul style="list-style-type: none"> - Trabajo en equipo
<p>Nombre: José Ramón</p> <p>Apellidos: Dongo</p> <p>Fecha de nacimiento: 03/03/1983</p> <p>Género: No binario</p> <p>Dirección:</p> <ul style="list-style-type: none"> - Tipo vía: Corredera - Vía: Ventolera - Número: 38 - Piso: Bajo - Puerta: B - Localidad: Toledo - CP: 45789 - Región: Castilla La Mancha

- País: España Período: - Fecha contrato: 15/06/2015 - Fecha cese: - Motivo: Jefe: Marta Padera
Nombre: Candelas Apellidos: Timero Fecha de nacimiento: 04/04/1984 Género: Otro
Nombre: Adela Apellidos: Drador Fecha de nacimiento: 05/05/1985 Género: Femenino
Nombre: Alberto Apellidos: Reón Fecha de nacimiento: 06/06/1986 Género: No binario
Nombre: José Ángel Apellidos: Atina Fecha de nacimiento: 07/07/1987 Género: Masculino
Nombre: Lurdes Apellidos: Enfreno Fecha de nacimiento: 08/08/1988 Género: Femenino
Nombre: Alfonso Apellidos: Najero Fecha de nacimiento: 09/09/1989 Género: Masculino
Nombre: Luis Gaspar Apellidos: Tidazo Fecha de nacimiento: 10/10/1990 Género: Masculino

Usuario

Nombre: jp.medario@corporacion.org Clave: 123456 Confirmación de clave: 123456 Fecha de última conexión:
Nombre: al.tica@corporacion.org Clave: altica93 Confirmación de clave: altica93 Fecha de última conexión:
Nombre: a.drador@corporacion.org

Clave: adela_d Confirmación de clave: adela_d Fecha de última conexión:
Nombre: ja.atina@@corporacion.org Clave: atina.ja.3 Conformación de clave: atina.ja.3 Fecha de última conexión:
Nombre: l.enfreno@corporacion.org Clave: sedrul Confirmación de clave: sedrul Fecha de última conexión:

Becarios

Nombre: Luisa Apellidos: Cacorchos Fecha de nacimiento: 01/01/1991 Género: Femenino Dirección personal: - Tipo vía: Plaza - Vía: Ficus - Número: 1 - Piso: 10 - Puerta: D - Localidad: Madrid - CP: 28098 - Región: Madrid - País: España Centro educativo: IES Sopor de verano Dirección centro educativo: - Tipo vía: Plaza - Vía: Ficus - Número: 1 - Piso: 10 - Puerta: D - Localidad: Madrid - CP: 28098 - Región: Madrid - País: España Mentor: Candelas Timero
--