

Entrega Persistencia

COMPETENCIAS

Esta entrega tiene por objetivo que el alumno sea capaz de:

- Diseñar e implementar el modelo de datos relacional que soporte la solución objetos planteada.
- Utilizar un ORM que permita implementar el modelo de datos previamente definido.

ENUNCIADO ESPECÍFICO

Usuarios

El sistema deberá persistir tanto los clientes como los administradores. Se deberán tener en cuenta todos los atributos mínimos requeridos en la entrega 0 para cada una de estas entidades.

Dispositivos

Los administradores son los únicos usuarios que podrán crear, modificar, o eliminar los dispositivos que soportará el sistema. Actualmente, los soportados son aquellos que se especifican en la sección “Tabla de dispositivos” de la entrega 2.

El sistema deberá persistir los dispositivos que utilizan los clientes junto con el consumo que genera cada uno de éstos. Es requerimiento del sistema que se tenga trazabilidad con respecto a los estados por los cuales pasó un dispositivo. Además, se deberá tener registro de aquellos dispositivos estándares que fueron convertidos a inteligentes por los usuarios.

Actuadores y sensores

El sistema deberá persistir cada uno de los sensores, junto con sus últimas N mediciones, así como también las reglas asociadas a cada uno de éstos y los actuadores que responden frente a cada aserción por regla.

Transformadores y geoposicionamiento

El sistema deberá persistir los transformadores que el ENRE nos envía mensualmente, así como también la posición geográfica en que se ubica cada uno de éstos.

Reportes

El sistema deberá mostrar los siguientes reportes con elevada frecuencia:

- Consumo por hogar/periodo.
- Consumo promedio por tipo de dispositivo (inteligente o estándar) por periodo.
- Consumo por transformador por periodo.

ENTREGABLES

Concepción y Comunicación del Diseño

- **Diagrama de clases actualizado**
- **Diagrama entidad relación (DER):** Este diagrama deberá especificar claramente:
 - Las claves primarias, las foráneas y las restricciones según corresponda.
 - Si fueran necesarias, estrategias de mapeo de herencia utilizadas. Se deberá adjuntar una justificación por cada estrategia de mapeo especificando por qué fue elegida, comparándola frente alguna otra.
 - Cómo fueron resueltos los impedance mismatches del modelo de objetos
- **Documentación de configuración general:** Documento que especifique cómo configurar la base de datos y como cargar datos de prueba en ella.

Implementación

Se deberá implementar el modelo de datos planteado como solución utilizando un ORM y una base de datos relacional. Además, se deberán implementar todos los casos de prueba que se plantean a continuación.

Casos de Prueba Mínimos

Caso de prueba 1:

Crear 1 usuario nuevo. Persistirlo. Recuperarlo, modificar la geolocalización y grabarlo. Recuperarlo y evaluar que el cambio se haya realizado.

Caso de prueba 2:

Recuperar un dispositivo. Mostrar por consola todos los intervalos que estuvo encendido durante el último mes. Modificar su nombre (o cualquier otro atributo editable) y grabarlo. Recuperarlo y evaluar que el nombre coincida con el esperado.

Caso de prueba 3:

Crear una nueva regla. Asociarla a un dispositivo. Agregar condiciones y acciones. Persistirla. Recuperarla y ejecutarla. Modificar alguna condición y persistirla. Recuperarla y evaluar que la condición modificada posea la última modificación.

Caso de prueba 4:

Recuperar todos los transformadores persistidos. Registrar la cantidad.

Agregar una instancia de Transformador al JSON de entradas. Ejecutar el método de lectura y persistencia. Evaluar que la cantidad actual sea la anterior + 1.

Caso de prueba 5:

Dado un hogar y un período, mostrar por consola (interfaz de comandos) el consumo total. Dado un dispositivo y un período, mostrar por consola su consumo promedio. Dado un transformador y un período, mostrar su consumo promedio. Recuperar un dispositivo asociado a un hogar de ese transformador e incrementar un 1000 % el consumo para ese período. Persistir el dispositivo. Nuevamente mostrar el consumo para ese transformador.

Recomendaciones

- Como motor de base de datos, se recomienda utilizar MySQL Server o MariaDB. Para las pruebas, puede resultar útil SQLite o H2.
- Como cliente de base de datos, se recomienda utilizar MySQL Workbench.
- Los ORM dependen del lenguaje de programación utilizado, por lo tanto se recomienda:

Java	<ul style="list-style-type: none"> • ORM: Hibernate • Documentación oficial: http://hibernate.org/orm/ • Ejemplo: https://github.com/dds-utn/jpa-proof-of-concept-template
C#	<ul style="list-style-type: none"> • ORM: EntityFramework • Documentación oficial: https://docs.microsoft.com/en-us/ef/core/get-started/aspnetcore/new-db • ORM: NHibernate • Documentación oficial: http://nhibernate.info/
PHP	<ul style="list-style-type: none"> • ORM: Doctrine • Documentación oficial: https://www.doctrine-project.org/

RESULTADOS DE APRENDIZAJE ESPECÍFICOS

- Comunicar las decisiones de diseño que se hayan tomado para cumplir con los requerimientos del sistema.
- Realizar el modelo de datos que responda al modelo de objetos planteado.
- Comunicar las estrategias de mapeo utilizadas en los casos necesarios.
- Configurar una base de datos relacional.
- Utilizar un ORM de mercado.