

Master ORM challenge

En la actualidad los **ORM** como **ActiveRecord** pueden ayudarnos a manejar data de manera facil y rápida. Nos permiten abstraernos de SQL y utilizar programación orientada a objetos para poder interactuar con nuestra data y representar nuestra lógica de negocio.

Los modelos es donde la magia pasa, responde a una convención *fat models*, y es que justo en ellos es donde declaramos como se comporta nuestra aplicación y cómo interactúan los datos. Son un gran aliado a la hora de definir comportamiento es por ello que para este proyecto tienen que considerarse las siguientes directivas:

- No importa cuantas migraciones tenga su proyecto, siempre que todas sean reversibles.
- Aplicar valores por default a las columnas de las tablas y/o constraints como not null
- Garantizar la integridad referencial mediante Foreign Keys
- Llenar data de prueba con seed/migrations
- Garantizar que todos los modelos planteados tengan las validaciones a nivel de modelos necesarias para asegurarse que solo data válida sea guardada en la base de datos.
- Utilizar al menos una validacion condicional coherente a la logica de negocios planteada
- Utilizar al menos un custom validation method
- Construir métodos de instancia que permita visualizar los errores friendly (json)
- Utilizar al menos un callback en la lógica de la aplicación.
- Usar callback relacionales donde apliquen
- Implementar el tipo de asociaciones necesarias para cumplir con el enunciado del problema, tener en cuenta conceptos aprendidos en el modelado de datos.

Enunciado

En el negocio marítimo existen empresas que se dedican a transportar mercancía de un punto a otro alrededor de todo el mundo, esto se hace en grandes buques, estas empresas se llaman **Navieras**, estos **buques** tiene sus nombres para identificarlos y pertenecen a estas grandes Navieras. Estos buques transportan mercancía en **containers** que son diferentes tamaños y tipos, cada container tiene un identificador único compuesto por una serie de caracteres alfanuméricos, los primeros cuatro caracteres son letras y los siguientes siete son números. Los principales tipos de containers son ST (Standard), HQ (HIGH-CUBE). Estos buques pueden cargar en promedio de 2000 a 8000 containers, es por ello que los importadores de los países no necesariamente pueden alquilar un buque completo para traer su mercancía, así es como entran en juego otras empresas llamadas NVOCC o **Consolidadores**, cuyo objetivo es agrupar un gran número Consignatarios (clientes finales) para que a las navieras les sea rentable ir de un lugar a otro con sus buques llenos. Los NVOCC pueden tener muchos clientes los cuales consiguen dando precios atractivos para los mismos, pero los clientes pueden también trabajar con muchos NVOCC ya que puede que para determinados orígenes obtengan mejores ofertas.

Con todo ese gran volumen de mercancía flotando por el mar, tiene que haber una manera de identificar a quién pertenecer la mercancía transportada, aquí entra en juego algo llamado **Bill of lading** (BL). El BL tiene toda la información necesaria del viaje que están realizando los containers, estos Bls son emitidos por las Navieras y en él podemos encontrar información como:

- Código del BL
- Nombre de la Naviera que transporta
- Nombre del NVOCC
- Nombre del Cliente Final
- Lista de todos los containers del cliente final
- Origen de la mercancía
- Destino de la mercancía
- Fecha de atraque
- Código de viaje (Este viaje es el que realizar el buque de un origen a destino)

A muy groso modo todo esto pasa todos los días alrededor del mundo, es por ello que se necesita representar esta lógica de negocios en un modelo de datos que cumpla los conceptos de *normalización* que permita obtener los siguientes reportes:

- En JSON mostrar la representación de la data de un BL
- Listar los BL por naviera
- Listar cuantos containers hay por tipos de containers
- Top 10 de países destino con más buques tocando puerto

El grupo de mentores evaluará:

1. **Estilo de codificación**
2. **Diseño de la solución al problema planteado**
3. **Funcionalidad**
4. **Cumplimiento de los objetivos planteados**

CONSIDERACIONES:

- El proyecto es estrictamente individual.
- Durante la realización del proyecto, surgirán dudas.
- No es recomendable quedarse con ellas. Platéelas a fin de aclararlas, preferiblemente por el canal de Slack “clases_y_retos”.
- El proyecto deberá ser entregado para el día Martes 20 Septiembre 2016 antes de las 9:00am.
- Cada estudiante tendrá 8 minutos como máximo para presentar su proyecto.
- La promo tendrá que coordinar ANTES del día de la presentación el orden en que presentarán los proyecto, de no haberse puesto de acuerdo, el tiempo de presentación se acortará a 5min por estudiante.

- La evaluación consistirá en presentar de forma lógica como se estructuró el proyecto, de la misma manera como interactúa la data entre los modelos y asociaciones.
- No se permite cargar data durante la presentación, la data debe de estar precargada.
- El mentor tendrá la potestad de al apreciar un mal funcionamiento del programa detener la presentación y pasar al siguiente estudiante.

BONO:

Si usted así lo desea y le queda tiempo suficiente, puede implementar los siguientes bonus points en su proyecto (no son de carácter obligatorio): * Usar Callback condicionales * Plantear en el problema un correcto uso de una asociación polimórfica. * Dos reportes fancy sobre la data en el sistema.