

# TTPS opción Ruby

## Trabajo integrador

Este documento define el alcance del trabajo integrador obligatorio con el cual los alumnos podrán obtener la cursada de la materia.

### El contexto

---

El estudio contable *E-Vader* te ha contactado para pedirte que desarrolles una aplicación web que les permita realizar algunas tareas que hoy por hoy llevan de forma manual, nada sistemática, en hojas de cálculo de *Open Office* y papel. Como te hicieron *una oferta que no podés rechazar*, aceptaste de muy buena gana realizar el desarrollo.

Están *muy* apurados por cambiar su forma de trabajo, y por eso te fijaron como fecha límite para tener la aplicación el día **Jueves 10 de Diciembre de 2015**. Ese día esperan tener toda la funcionalidad implementada y lista para ser usada, aunque pueden aceptar que la interfaz de usuario no sea *muy linda*.

A continuación tenés el resultado del análisis de requerimientos que tu Product Owner obtuvo tras algunas reuniones con los futuros usuarios de la aplicación, depuradas a un lenguaje relativamente técnico.

### Los requerimientos

---

Se te pide que desarrolles una aplicación para llevar el libro de facturación de los clientes del estudio.

Cada cliente debe tener:

- Datos personales (nombre, apellido, fecha de nacimiento, género, número de documento, CUIL/CUIT)
- Diferentes medios de contacto (teléfono, email, skype, dirección postal, y/o los que te parezcan adecuados)
- facturas emitidas (donde   $\geq 0$  )

Una factura, a su vez, debe poseer:

- Un cliente del estudio que la emite
- Datos de la persona (física o jurídica) a la que fue emitida (nombre o razón social, CUIL/CUIT)
- Una descripción de los bienes o servicios por los cuales fue emitida
- Un monto total siempre expresado en Pesos
- Una fecha de emisión

Queda a tu criterio interpretar qué atributos son requeridos en cada caso, reflejando esto en validaciones del modelo de datos y en restricciones de la base de datos (*columnas* `NOT NULL` ).

Sobre la interfaz de usuario, han pedido que se les brinden los módulos necesarios para gestionar toda esa información de una forma cómoda y sencilla, poniendo énfasis en poder ver dentro de la información detallada de un cliente los siguientes ítems:

- La edad del cliente.
- El monto total facturado por año para el cliente (al estilo *"En 2015 facturó \$ 999.999,00"*).
- La cantidad de facturas que el cliente ha emitido por mes desde Enero del corriente año.
- Las 5 personas a las que más les ha facturado ese cliente.

## Las restricciones

---

La aplicación debe:

- Ser versionada utilizando Git.
- Estar implementada en el framework Rails.
- Persistir los datos en alguna base de datos relacional (SQLite, MySQL, *choose your poison*).
- Tener tests de unidad, al menos, para las siguientes situaciones:
  - La creación de un nuevo cliente:
    - Sin datos de contacto
    - Con datos de contacto
    - Con algún dato personal que no sea válido
  - La carga de una nueva factura emitida por un cliente:
    - Sin errores de validación
    - Con errores de validación
  - El borrado de un cliente:
    - Sin facturación registrada
    - Con alguna factura emitida
- Tener documentación de cómo preparar el ambiente para su correcto funcionamiento.
- Tener documentación de cómo ejecutarla.
- Tener un set de datos inicial ( `seeds` ) que contenga cualquier dato básico para su funcionamiento y un conjunto de clientes con facturación ya cargados para poder realizar pruebas.

La entrega debe:

- Realizarse de manera individual.
- Hacerse enviando el link al repositorio del proyecto en GitHub.
- Contener toda la información y documentación necesaria para ponerla en funcionamiento.
- Cumplir, al menos, con todas las necesidades detalladas en la sección **Los requerimientos**.
- Definir y pasar todos los tests de unidad pedidos.

