

Trabajo final integrador

TTPS - Opción Ruby - Cursada 2017

Este documento define el alcance del trabajo integrador obligatorio con el cual l@s estudiantes podrán obtener la cursada de la materia.

El contexto

Nuestra cátedra necesita un sistema para gestionar las notas de l@s estudiantes a lo largo de las distintas instancias de evaluación que definimos, en los distintos años de cursada que vamos dictando.

Cada año, se realizan evaluaciones parciales - cuya cantidad, fechas y nota mínima de aprobación pueden variar - y un trabajo final integrador - como este que estás leyendo -, que se definen ese mismo año.

Con este sistema se busca almacenar en un único lugar los resultados de esas evaluaciones para cada estudiante, permitiéndonos determinar fácilmente qué notas obtuvo en cada instancia de evaluación y si aprobó o no la cursada.

A continuación tenés la lista de requerimientos que desde la cátedra hemos definido para la implementación de la aplicación, y luego vas a encontrar algunas restricciones sobre cómo implementarla y qué tecnologías utilizar.

Los requerimientos

Se te pide que desarrolles una aplicación web para llevar registro de las notas de las evaluaciones de diferentes cursadas de esta materia.

La aplicación será accedida por los docentes de la cátedra, que deberán identificarse mediante alguna forma de autenticación¹. Si quien accede a la aplicación no se encuentra debidamente identificado, no podrá acceder a ninguna parte de la misma.

La materia maneja distintas instancias de evaluación para cada año (o *cursada*, como solemos llamarlos), que se definen en el mismo año y tienen características particulares a cada una de esas instancias: un título descriptivo, la fecha en que se realiza la evaluación y la nota mínima de aprobación.

Además de esto, se deben poder gestionar estudiantes inscript@s a la materia por año. Cada año l@s estudiantes inscript@s son incorporad@s al sistema y podrán relacionarse con las instancias de evaluación de ese año. Por motivos de simplicidad, podés considerar que si alguien recursa la materia, se volverá a cargar en el segundo año que la curse (como si fuera otr@ estudiante). De cada estudiante se espera tener, como mínimo, los siguientes datos: apellidos, nombres, DNI, número de legajo y correo electrónico.

Las instancias de evaluación deben poder definirse primero, y luego, como una operación posterior en el tiempo, se deben poder cargar las notas de quell@s estudiantes que hayan rendido. Quienes no tengan nota cargada, tendrán un “Ausente” por defecto.

Como en la cátedra somos informáticos, personas normales, pero tenemos determinados códigos de estética visual, esperamos que la aplicación sea cómoda de utilizar y que utilice

¹ Es tu responsabilidad investigar qué gemas existentes te proveen algún mecanismo de autenticación para integrar en el proyecto y elegir una para utilizarla en la aplicación.

alguna librería² de componentes o de estilos visuales para estandarizar cómo se verán y presentarán las distintas páginas que la componen.

Al utilizar la aplicación, se espera contar con la posibilidad de realizar las siguientes operaciones:

- Gestión (CRUD) de años de cursada.
- Gestión (CRUD) de instancias de evaluación dentro de un año de cursada.
- Gestión (CRUD) de estudiantes inscript@s en un año de cursada.
- Carga y modificación de las notas de l@s estudiantes en una evaluación en particular.
- Visualización en una sola página de todos los resultados de l@s estudiantes en las distintas evaluaciones de un año de cursada, ordenados por fecha de evaluación.
 - Esta vista debe consignar, junto a cada nota, si el/la estudiante está aprobad@ (excepto que no se haya presentado).
 - Adicionalmente, debe presentar un resumen, por evaluación, de los resultados con la siguiente información:
 - Cantidad de estudiantes que han aprobado.
 - Cantidad de estudiantes que han desaprobado.
 - Cantidad de estudiantes ausentes.
 - Porcentaje de aprobación (número de aprobados sobre el total de alumnos que se presentaron).

Las restricciones

La aplicación debe:

- Ser versionada utilizando **Git**.
- Estar implementada en el framework **Ruby on Rails** en su última versión estable (al momento de publicación de este trabajo es **5.1.4**).
- Persistir los datos en alguna **base de datos relacional** (SQLite, MySQL, *choose your poison*).
 - Queda a tu criterio interpretar qué atributos del modelo de datos son requeridos en cada caso, reflejando esto en validaciones del modelo de datos, valores por defecto y restricciones de la base de datos.
- Utilizar alguna **gema de autenticación**. A continuación se muestran algunos ejemplos:
 - [Devise](#)
 - [Sorcery](#)
 - [Authlogic](#)
 - [Shield](#)
- Tener al menos 15 **tests de unidad de relevancia** para el modelo de datos, sus inter-relaciones y la lógica de calificación de las evaluaciones (poder *preguntarle* a una evaluación si un/a estudiante aprobó o no, por ejemplo).
- Tener una **documentación escrita** que al menos consigne:

² Las opciones más comunes pueden ser: [Bootstrap 4](#), [Foundation 6](#) o [Bulma CSS](#), entre otros. *Choose your poison*. Con esto apuntamos a que se mejore la estética general de la aplicación, pero jamás pretendemos que esto insuma más de un 10% del tiempo que se dedique al trabajo.

- Cómo preparar el ambiente para el correcto funcionamiento de la aplicación (dependencias, requerimientos en el ambiente en que se va a correr la aplicación, preparación de la base de datos, etcétera).
- Cómo ejecutar la aplicación.
- Cómo ejecutar los tests.
- Tener un **set de datos inicial** (*seeds*) que contenga cualquier dato básico para su funcionamiento. Como mínimo, este set inicial deberá proveer:
 - Dos cursadas (2016 y 2017).
 - Diferentes instancias de evaluación para cada cursada.
 - Al menos 10 estudiantes, los cuales deberán estar en cursada(s) y tener notas cargadas en evaluaciones.

La entrega debe:

- Realizarse de manera **individual**.
- Hacerse **enviando la siguiente información**:
 - El link al repositorio del proyecto en el hosting de Git que desees utilizar³ (GitHub, BitBucket, GitLab, etcétera).
 - El link a la aplicación corriendo en Heroku (este último, según corresponda⁴).
- Contener toda la información y documentación necesaria para ponerla en funcionamiento en un archivo **README.md** en el directorio raíz del repositorio.
- **Cumplir como mínimo con todos los requerimientos** detallados en la sección *Los requerimientos*.
- Definir, implementar y pasar exitosamente con todos los **tests de unidad** pedidos.

³ Tené en cuenta que si vas a utilizar un repositorio privado, los integrantes de la cátedra necesitaremos acceso al mismo, para lo cual necesitás escribir a ncuesta@info.unlp.edu.ar indicando el hosting que vas a utilizar, para coordinar los datos de los integrantes que debés autorizar.

⁴ El despliegue de la aplicación en **Heroku** será opcional en la primera fecha de entrega y será **obligatorio en la segunda fecha de entrega**.