

Antecedentes

Actualmente, la ESPOL se encuentra realizando un proyecto institucional para que todos sus estudiantes *aprendan a programar*.

En un inicio se creía que la programación era exclusiva de los informáticos, luego se dio paso a que algunas carreras de ingeniería lo aprendan, hasta que finalmente el tiempo ha demostrado que [todo el mundo debería saber programar](#).

Sin embargo, aprender a programar no es algo sencillo, ya que es una habilidad que se va desarrollando con la práctica. En muchos casos, hay estudiantes que captan la lógica de programación bastante rápido, y en cambio hay otros que les cuesta un poco más es por esto que se requiere de una herramienta que contribuya al aprendizaje de esta materia.

Objetivos

- Creación del sitio institucional de la materia de Fundamentos de Programación.
- Creación de una herramienta educativa (**Sandbox**) que utilice elementos de gamificación y permita a los estudiantes practicar ejercicios de programación.

Descripción

Sitio Institucional

En el proyecto parcial se les solicitó la creación del sitio institucional de la materia de Fundamentos de Programación con desarrollo únicamente del lado del cliente. Sin embargo, para esta ocasión se le solicita implementar el sitio utilizando uno de los frameworks más populares del mundo: [Python Django](#).

Para el desarrollo del sitio institucional no es necesario implementarlo con AJAX y APIs puede utilizar el modelo tradicional de devolver una página y embeber código python para mostrar el contenido. Solo existe un tipo de usuario para el sistema que es el administrador y que básicamente puede actualizar todo el contenido del sitio.

El sitio contará con los siguientes elementos:

- Página de Inicio
- El curso: descripción, requisitos, syllabus y políticas del curso. Es obligatorio que esta información se encuentre en una base de datos, no puede colocar solo archivos PDFs

- Semana a semana. Es obligatorio embeber una de las clases dictadas que se encuentra en el canal de youtube de espol:
https://www.youtube.com/playlist?list=PLKoPE99M_ulU1y0Zb_8zX_5lgK6OLXtzM
- Equipo
- Ayudantías
- **(sección nueva)** Noticias: se dispondrá de una sección de noticias que permita el ingreso de noticias y que puedan ser visualizadas por cualquiera.
- **(sección nueva)** Rankings: se dispondrá de una sección ranking que realizará una consulta al API del sandbox para obtener los estudiantes con mayor puntaje de cada curso de la materia.

Sandbox

En el proyecto parcial se le solicitó la creación de una herramienta que permita realizar ejercicios de python que pertenecía al mismo sitio de la materia. Para esta ocasión usted realizará una aplicación aparte con el sandbox. Esta aplicación será desarrollada enteramente con node, express, mongodb y angular.

Usuarios

Dentro de la aplicación del sandbox existirán 4 tipos de usuarios:

- Administrador
- Profesor
- Ayudante
- Estudiante

Funcionalidades

Login del sistema

El login al sistema será a través del correo electrónico y una contraseña y deberá ser realizado con la librería de passport que provee una excelente gestión para la autenticación:

<http://passportjs.org/>

Módulo de Usuarios

Este módulo se hará cargo de la gestión de los usuarios del sistema. *Es de uso exclusivo del administrador.* Para todo usuario del sistema se contará con:

1. Correo Electrónico
2. Contraseña
3. Rol
4. Tipo de identificación (cédula en el caso profesores o matrícula en el caso de estudiantes)
5. Identificación
6. Nombres
7. Apellidos
8. Carrera (en el caso de los estudiantes)

Este módulo debe proveer de un CRUD para toda la administración y se deberá poder ordenar y filtrar de acuerdo a los atributos previamente mencionados. Para esta última parte usted puede hacer uso de la herramienta datatables que adicionalmente cuenta con una integración con bootstrap para que la tabla tenga el mismo estilo de este framework:

<https://datatables.net/>

<https://datatables.net/manual/styling/bootstrap>

Al crear un usuario, el sistema deberá enviar un correo electrónico al usuario con una contraseña temporal la cual puede ser cambiada. Para el envío de correos electrónicos revisar:

<https://community.nodemailer.com/>

Módulo de Cursos

Este módulo se hará cargo de la gestión de cursos del sistema. Es de uso exclusivo del administrador. Un curso está compuesto de:

1. Profesor
2. Número de Paralelo
3. Lista de estudiantes

Para la creación de un curso usted deberá implementarlo utilizando la funcionalidad de autocomplementar que permita seleccionar a un profesor y a los estudiantes. Asimismo deberá ser posible actualizar y eliminar un curso.

Debe ser posible visualizar todos los estudiantes de un curso ya sea a través del Profesor y/o número de paralelo. Recuerde es posible que un profesor tenga dos paralelos.

Finalmente, el sistema también proveerá la creación de estudiantes y cursos mediante la subida de un archivo CSV que contenga la información de los estudiantes y del curso.

Módulo de Ejercicios

Este es el módulo principal del sandbox y tiene funcionalidades de acuerdo al rol del usuario:

Si el usuario es un profesor o un ayudante, podrá gestionar la creación de ejercicios (CRUD). Un ejercicio está compuesto de:

- Título
- Descripción
- Datos de entrada para validar que funcione
- Datos de salida para validar que funcione
- Un conjunto de etiquetas
- Nivel de dificultad (fácil, intermedio, difícil)

Además, podrá visualizar todos los ejercicios creados en el sistema.

Si el usuario es un estudiante, podrá resolver ejercicios.

Para resolver un ejercicio el estudiante primero deberá seleccionar uno de los tres niveles de dificultad y un tag. Luego le aparecerá aleatoriamente un ejercicio que este en esa dificultad y tenga ese tag. Si el estudiante no quiere realizar ese ejercicio puede escoger una opción de saltar que en cambio lo llevara a otro ejercicio.

Es necesario que el sistema valide la solución del estudiante. Por lo tanto, el sistema deberá correr el código subido por el estudiante con los datos de entrada del ejercicio y deberá validarlos con los datos de salida.

Existen algunas maneras de llevar a cabo esta parte. Una alternativa es utilizar la librería *python-shell*:
<https://github.com/extrabacon/python-shell>

Otra alternativa es realizarlo mediante el uso de child processes (de hecho, la primera alternativa lo hace así), en esta alternativa usted lanzara un proceso hijo que ejecutara python sobre el sistema operativo. Puede leer mas en el siguiente link:

<http://www.sohamkamani.com/blog/2015/08/21/python-nodejs-comm/>
<http://stackoverflow.com/questions/34213845/call-python-script-using-node-js-child-process>

Cuando un estudiante resuelve exitosamente un ejercicio se le otorga un puntaje de acuerdo a la complejidad:

Fácil (5 puntos)

Intermedio (10 puntos)

Difícil (15 puntos)

También cuando un estudiante haya resuelto cierta cantidad de ejercicios se le otorgará una insignia (badge). De acuerdo a lo siguiente:

10 ejercicios: Insignia Novato

20 ejercicios: Insignia PRO

30 ejercicios: Insignia Experto

5 ejercicios en una semana: Insignia Indestructible

10 ejercicios en una semana: Insignia Duro de Matar

15 ejercicios en espacio de una semana: Insignia Rápidos y Furiosos

Módulo de Desafíos (Mejoramiento)

Usted creara un modulo de desafíos para que los estudiantes compitan por la obtención de una insignia especial.

Si el usuario es un profesor, podrá crear un desafío que es básicamente seleccionar hasta 5 ejercicios de todos los que están en el sistema. Además, tendrá que ingresar un título y un tiempo máximo para realizarlo. Y debe permitir eliminar un desafío.

Si el usuario es un estudiante, podrá participar en el desafío creado por su profesor (solo pueden participar los del curso del profesor). Un estudiante podrá visualizar todos los ejercicios del desafío y podrá seleccionar por cual comenzar. Usted deberá mostrar el tiempo que le queda para terminar el desafío. Una vez concluido el tiempo, el sistema deberá redireccionar al usuario a una pagina con los resultados (quien fue el estudiante ganador).

Esta ultima parte tendrá que ser implementada con tecnologías de "tiempo real" y usted puede realizarlo ya sea utilizando el servicio de pusher (tiene un free plan):

https://pusher.com/docs/javascript_quick_start

<https://github.com/pusher/pusher-http-node>

O (para los aventureros) con websockets:

https://www.tutorialspoint.com/html5/html5_websocket.htm

<https://github.com/theturtle32/WebSocket-Node>

Nota: No puede utilizar long polling o setTimeout

El ganador se determina de acuerdo al numero de ejercicios realizados y al tiempo que le tomo en resolverlos. El ganador se hace acreedor a una insignia especial que esta determinada de acuerdo a lo siguiente: Si el profesor ha publicado entre 0 y 5 ejercicios, la insignia especial es de bronce; entre 6 y 10 ejercicios, es de plata; mas de 10 ejercicios, es de oro.

Módulo de Reportes

Usted deberá proveer de un pequeño modulo de reportes que le permitan a un administrador conocer ciertas métricas de la aplicación.

La aplicación deberá mostrar:

- Un gráfico que muestre la cantidad de ejercicios resueltos por día dentro de un rango especificado por el usuario.
- Un grafico que muestre la cantidad de ejercicios resueltos por curso.

Adicionalmente, usted deberá proveer de un API para consultar los 3 mejores estudiantes de cada curso.

Mi Perfil

Los estudiantes contarán con una pagina de perfil en donde se visualizara el nombre completo del estudiante, el puntaje obtenido, las insignias obtenidas, y el numero de ejercicios completados.

Entregables

El proyecto deberá ser realizado en las maquinas virtuales provistas por el CTI. Asimismo, deberá tener un repositorio en Github para cada aplicación. Es obligatorio que realicen el ingreso de las tareas a realizar por cada estudiante a través de la aplicación provista por el CTI.