Propuesta de investigación Primavera 2018

Leonardo Gamaliel Velarde Andrade

Introducción

Crear un sistema modular de Deep learning consumible vía un servicio RestFull. Mandando información tanto de aprendizaje y de muestreo y retornando regresiones o, en dado caso, datos configurables vía json o, en casos específicos, XML.

Requerimientos

El sistema requiere hacer un servicio RestFull, tal como lo ofrece Sails.js: un framework de Node.js para crear, gestionar y optimizar Apis de una forma sencilla y rápida.

El sistema va a requerir interfaces de usuario y experiencia de usuario (UI, UX) para crear un sistema amigable, sencillo y moderno para que los usuarios que utilicen la plataforma puedan optimizar sin la necesidad de saber de computación, su cuenta y su propio servicio.

También se va a requerir trabajar redes neuronales para Deep learning, usando Julia o TensorFlow (Frameworks) o, en dado caso, crear un propio framework que pueda trabajarse de forma modular.

**Recomendaciones y enlaces

Se recomienda trabajar con Sails.js [1] en Node.js[2], con una imagen en docker[3] y un repositorio en github[4]; optimizando el desarrollo grupal y creando un ecosistema ad hoc a las máquinas de producción, escalando de una manera increíblemente mayor y preparando el sistema para hacer devOps[5] después de la primera versión lanzada a producción.

Para el backEnd se prevee utilizar Python[6] para el procesamiento y la implementación del Deep Learning[7] y MongoDB[8] para crear una base de datos no SQL por las necesidades que el proyecto requiere.

Sails.js

Es un framework que corre en el lado del servidor desde Node.js. Esta herramienta cuenta con una incorporación nativa con Socket.io[9] lista para ser consumible vía un librería que también ya contiene Sails.js que corre en el lado del cliente para tiempo real. Sails usa como motor a Express.js otro framework reconocido por tener una comunidad amplia y sobre documentación. Además de contener un ORM excepcional llamado Waterline [10] que nos ayuda a poder trabajar con cualquier base de datos.

[1] - https://sailsjs.com/ - Página oficial donde se podrá encontrar documentación completa, estructura del framework, ejemplos de código y la forma óptima de descargar e instalarlo.

[2] - https://www.w3schools.com/nodejs/nodejs_intro.asp - Página dedicada a describir el funcionamiento óptimo de Node.js desde w3shools.

[3] - https://www.redhat.com/es/topics/containers/what-is-docker - Breve introducción a Docker por la comunidad de redHat en español.

[4] - http://tombatossals.github.io/git-puesto-en-practica/trabajando-en-equipo/ - ¿Qué es y por qué ocupar git para equipos de trabajo?

[5] - https://www.paradigmadigital.com/techbiz/que-es-devops-y-sobre-todo-que-no-es-devops/ - ¿Qué es y qué no es DevOps?

[6] - https://www.python.org/ - Página oficial de Python, ejemplos, descargas y breve introducción al trabajo con Python.

[7] - https://www.xataka.com/robotica-e-ia/deep-learning-que-es-y-por-que-va-a-ser-una-tecnologia-clave-en-el-futuro-de-la-inteligencia-artificial - Deep Learning: qué es y por qué va a ser una tecnología clave en el futuro de la inteligencia artificial.

[8] - https://www.genbetadev.com/bases-de-datos/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no - ¿Qué es MongoDb? Breve introducción y comparación de Mongo DB vs SQL.

[9] - https://carlosazaustre.es/websockets-como-utilizar-socket-io-en-tu-aplicacion-web/ - Breve tutorial y explicación de WebSockets con Socket.io

[10] - https://sailsjs.com/documentation/concepts/models-and-orm - Conceptos, usos y documentación completa y oficial para el ORM por defecto de Sails.js

[**] - https://www.sohamkamani.com/blog/2015/08/21/python-nodejs-comm/ - La comunicación entre Python y Node.js de forma nativa con *child_process,* creando un hilo independiente y asíncrono.

Historias de usuarios

CONFIGURACIÓN

El usuario *Empresa* visita la página principal, donde encontrará una pequeña landing page describiendo el servicio que se ofrece: Una imagen descriptiva, logo, misión, visión y estadísticas que responden a las preguntas: ¿Qué y por qué nosotros? Se ve, además, planes, costos y ejemplos de desarrollo. Finalmente encontrará un bloque de registro destacado, donde podrá darse de alta o como usuario particular o como Empresa. Habrá un texto remarcado para poder elegir entre ambos usuarios: "*El usuario particular o desarrollador podrá gestar proyectos y traerlos desde github y el usuario Empresa podrá crear usuarios desarrolladores vinculados a su empresa, con lo cual podrá dar seguimiento*". El usuario *Empresa* se dará de alta como Empresa, y una vez confirmado su correo y su número de celular, entrará a la plataforma y verá una pasarela donde, por medio de pequeñas cards informativas observará sus proyectos. La card informativa tendrá título del proyecto, espacio utilizado, última facturación (en caso que sea usuario Premium) y tokens del proyecto. Después de la pasarela *Empresa* verá un botón "Crear nuevo proyecto", donde, si es un usuario con cuenta activa, podrá clickar y generar un nuevo proyecto. El botón, inmediatamente después de ser presionado, abrirá un formulario donde *Empresa* escribirá los metadatos del proyecto: "Nombre, Locación de su proyecto, parámetros aceptados y parámetros devueltos". Lo siguiente que Empresa verá, es un formulario para crear colecciones, ejemplos, y nombrará a los parámetros devueltos y aceptados. Finalmente, al terminar este formulario, encontrará la documentación con cURL para hacer post o get a su respectivo EndPoint, junto con las variables token, donde encontrará el token generado para el proyecto, y el nombre (en un esquema de json) para el envío de variables y un breve ejemplo de cURL y la respuesta de la API.

SISTEMA

El usuario Sistema deberá llevar la información de evaluación vía post o get al EndPoint registrado previamente, junto con su respectivo token. El API evalúa esa información y el Sistema recibirá la respuesta para el procesamiento que requiera hacer vía websockets al cliente en cuestión.

SUPERVISOR/ANALISTA

El usuario Analista deberá estar ligado a una cuenta Empresarial, donde, después de iniciar sesión (con una sesión creada y ligada vía invitación Empresa – usuario), podrá entrar a la estadística de cada proyecto mantenido por el usuario Empresa. Tal estadística estará dispuesta empezando por métricas de configuración, representadas en un pequeo boque en la parte de arriba de la página donde verá los ejemplos y la configuración proyecto que, claramente, no podrá modificar más que la empresa o el Owner. La estadística estará en tiempo real, viendo los *response* y los *request* en una pequeña tabla que se actualiza

con el código de response (200, 400, 500, …) y, finalmente, verá las gráficas que podrá o globalizar por toda la información o crear subconjuntos lógicos enviando scripts en Python con una librería diseñada para estadística, dentro de una pequeña terminal en la parte inferior de la gráfica.

ADMINISTRADOR

El usuario Administrador es un usuario empresa que tendrá acceso a una app para supervisar facturación y ver estadística general, cambios mínimos en los parámetros de los proyectos y podrá enviar invitaciones para crear usuarios Analistas.