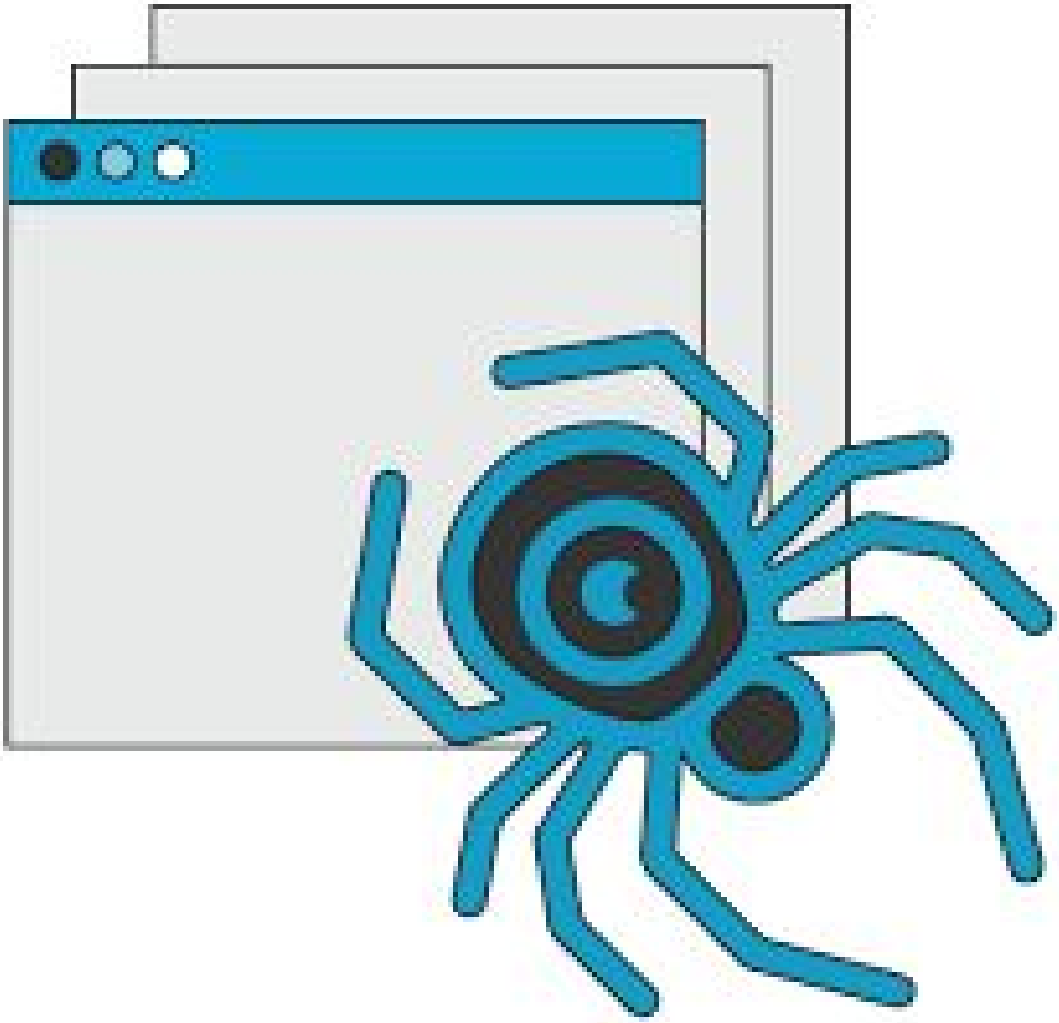
08/12/2020

Curs 2020-2021



Documentació corresponent al treball de scrapping sobre la nostra pròpia pàgina web

Mateo Garcia y Adrià Flexas
Desenvolupament d'aplicacions WEB, IES FB MOLL

Index

Index de ilustració

Descripció general del projecte

En aquest projecte, la nostra tasca serà la creació d'un lloc web utilitzant únicament HTML y CSS, després de realitzar aquesta tasca, el nostre objectiu serà la construcció d'un crawler que es dedicara el cercar els links de la nostra pàgina y després la construcció d'un scrapper que extraura l'informació de dintre d'aquests links per a finalment convertir aquestes dades a documents json y conseguir guardar-los a una base de dades(mongo)

Requeriments necessaris del projecte per a cada mòdul/assignatura:

Sistemas Informáticos:

- Instala desde línea de comandos (CLI) todo el software necesario para desarrollar el proyecto, tanto para la escritura de código Python, como HTML5, CSS3, y el sistema gestor de bases de datos.
- Instala desde CLI Python 3.X.
- Invoca desde línea de comandos el programa Python y averigua qué parámetros admite.
- Averigua qué errores genera una mala invocación del programa.
- Instala desde CLI el entorno de desarrollo, sus extensiones y las herramientas de testing y debugging.
- Instala el gestor de paquetes pip desde línea de comandos.
- Instala las librerías necesarias de Python con el gesto de dependencias pip.
- Instala desde línea de comandos (CLI) el control de versiones git.
- Utiliza desde CLI el control de versiones.

- Crea desde línea de comandos en tu máquina un directorio donde almacenar los ficheros con los documentos JSON. Nos referiremos a este directorio como la biblioteca de documentos.
- Copia o mueve desde línea de comandos los documentos de la biblioteca.
- Ejecuta el programa desde línea de comandos para extraer determinada información del sitio web sobre el que realizas el scraping.
- Instala desde CLI MongoDB y Compass y configura el motor de BBDD.
- Arranca desde línea de comandos MongoDB y Compass y realiza operaciones CRUD sobre la base de datos.
- Arranca desde línea de comandos Compass para estudiar el Esquema de los documentos de la base de datos.

Bases de datos

- Realiza el modelo del esquema de la base de datos orientada a documento que almacenará los JSON resultado del scraping del sitio web.
- Crea una colección de documentos JSON con los productos y servicios resultado del scraping del sitio web. La especificación del esquema de los documentos ha de ser pactada con el equipo de segundo de Dual que consumirá tus JSON para extender el negocio. Si no se cumple el requisito, el proyecto no se considera resuelto satisfactoriamente.
- Define el tipo de dato de cada campo del documento.
- Crea una base de datos en la nube mediante el servicio Atlas de MongoDB.
- Define la autenticación y la autorización sobre la base de datos.
- Define las colecciones de documentos que estimes oportunas.

- Utiliza la utilidad Compass de MongoDB para:
    - Consultar la información almacenada en la base de datos.
    - Visualizar las estadísticas sobre uso, presencia, rangos de valores, tipos y demás parámetros que ofrece Compass.
- Utiliza el lenguaje de manipulación de datos de MongoDB para:
    - Consultar la información almacenada en la base de datos.
    - Modificar la información almacenada en la base de datos.

Lenguaje de Marcas

- La profesora del módulo te asignará uno de los componentes (reserva de UFOs, reserva de packs de bienvenida o reserva de menús) que habrás de construir.
- Crea el sitio web corporativo de la organización que te hayan asignado. Invéntate los productos y servicios hasta conseguir un sitio funcional y representativo del negocio.
- Recuerda que el esquema de los documentos ha de ser pactado con el equipo de segundo de Dual que consumirá tus JSON para extender el negocio.
- Utiliza las etiquetas semánticas que ofrece HTML5 para etiquetar la información de los productos y servicios que ofrece el negocio.
- Utiliza CSS3 para garantizar una buena experiencia de usuario y usabilidad del sitio.
- Utiliza HTML5 y CSS3 para garantizar la accesibilidad del sitio.
- Valida tu código HTML5 y CSS3 con los servicios de validación que ofrece W3C.
- Utiliza el esquema que has propuesto en el módulo de bases de datos para definir los elementos HTML que etiquetarán la información sobre los productos y servicios.

- Define los espacios de nombres que sean necesarios.

Entornos de Desarrollo

- Crea un repositorio del proyecto en GitHub.
- Crea un fichero README.MD que describa adecuadamente el proyecto. Utiliza Markdown para preparar documentación ágil sobre el proyecto.
- Realiza el control de versiones y trabaja de manera colaborativa con tu pareja de programación.
- Utiliza ramas para realizar los evolutivos, testing y debugging, siguiendo el flujo de trabajo que especificaremos en las sesiones presenciales.
- Documenta todo el proceso.
- Escribe las historias de usuario/a = identifica y describe los requisitos funcionales de la aplicación.
- Elige el ciclo de desarrollo que consideres más adecuado a tu manera de trabajar (y a la de tu pareja de programación) y al proyecto.
- Utiliza Clockify para llevar un seguimiento del tiempo de desarrollo empleado en cada una de las fases del ciclo de vida del proyecto. Sé riguroso porque los informes generados por esta herramienta serán un artefacto a entregar en la defensa del proyecto.

Programación

- Escribe los componentes de Python que extraigan los datos del negocio de los sitios web -sus productos y servicios- y que genere un sistema un sistema de almacenamiento en formato JSON para estos datos extraídos.
- Divide el código en rutinas de modo que sea SRP y OCP.
    - SRP (S) o Principio de Única Responsabilidad o Single Responsibility Principle. Un componente sólo debe tener un motivo para cambiar.
    - OCP (O) o Open/Closed Principle. Las entidades de software (clases, módulos, funciones, etc.) deben estar “abiertas” a la extensión pero “cerradas” a la modificación.
- Emplea precondiciones y postcondiciones para chequear las invariantes (las estructuras de datos) que maneja el programa y que intercambian las distintas rutinas y módulos que componen el sistema.
- No puedes utilizar las librerías disponibles en lenguaje Python para parsear el DOM o hacer scraping del sitio web. Puedes observar las funcionalidades que ofrecen, estudiar su código y la lógica de la aplicación, pero si empleas alguna de ellas, el proyecto no se dará por válido.
- Elige la estructura de datos adecuada para representar y manipular en memoria los documentos.
- Utiliza bloques try /except para capturar las excepciones que se puedan producir durante la ejecución del programa.
- Crea una colección de documentos JSON con los productos y servicios resultado del scraping del sitio web. La especificación del esquema de los documentos ha de ser pactada con el equipo de segundo de Dual que consumirá tus JSON para extender el negocio. Si no se cumple el requisito, el proyecto no se considera resuelto satisfactoriamente.
- Invoca desde consola el programa y pásale las opciones adecuadas.

Procediments de desenvolupament

Eines utilitzades:

Hem utilitzat el IDE Visual Studio Code per a desenvolupar tot el codi, amb el Visual Studio Code hem utilitzat altres eines com un linter per a poder veure les errades que cometem, també hem fet ús de l'extensió "Conventional Commits" que ajuda a fer els commits d'acord als conventional commits. També hem fet ús de Git Graph, aquest mostra un esquema dels branches, a més de dur un historial amb tots el commits fets.

Per a poder treballar junts d'una manera més eficient hem fet ús del "Live share", un altre extensió que permet treballar al IDE d'un altre persona al mateix temps.

Planificació:

Hem fet ús de la metodología en V, a més de posar-nos a nosaltres mateixos una sèrie de mini-sprints.

La metodología en V es una mena de metodología en cascada per amb la diferència de que no es necessari acabar imperativament el mòdul abans de poder continuar treballant en el següent, és a dir dóna la possibilitat de tornar enrera.

Al principi del projecte ens varem centrar en organitzar-nos per ser més productius y finalment varem decidir que utilitzem els sprints, és a dir per a cada setmana ens posavem uns objectius per a fer que normalment assolíem.

A la vegada, al principi del projecte estavem un poc perduts però gracis als sprints ens varem acabar organitzant bé.

Requisits no funcionals:

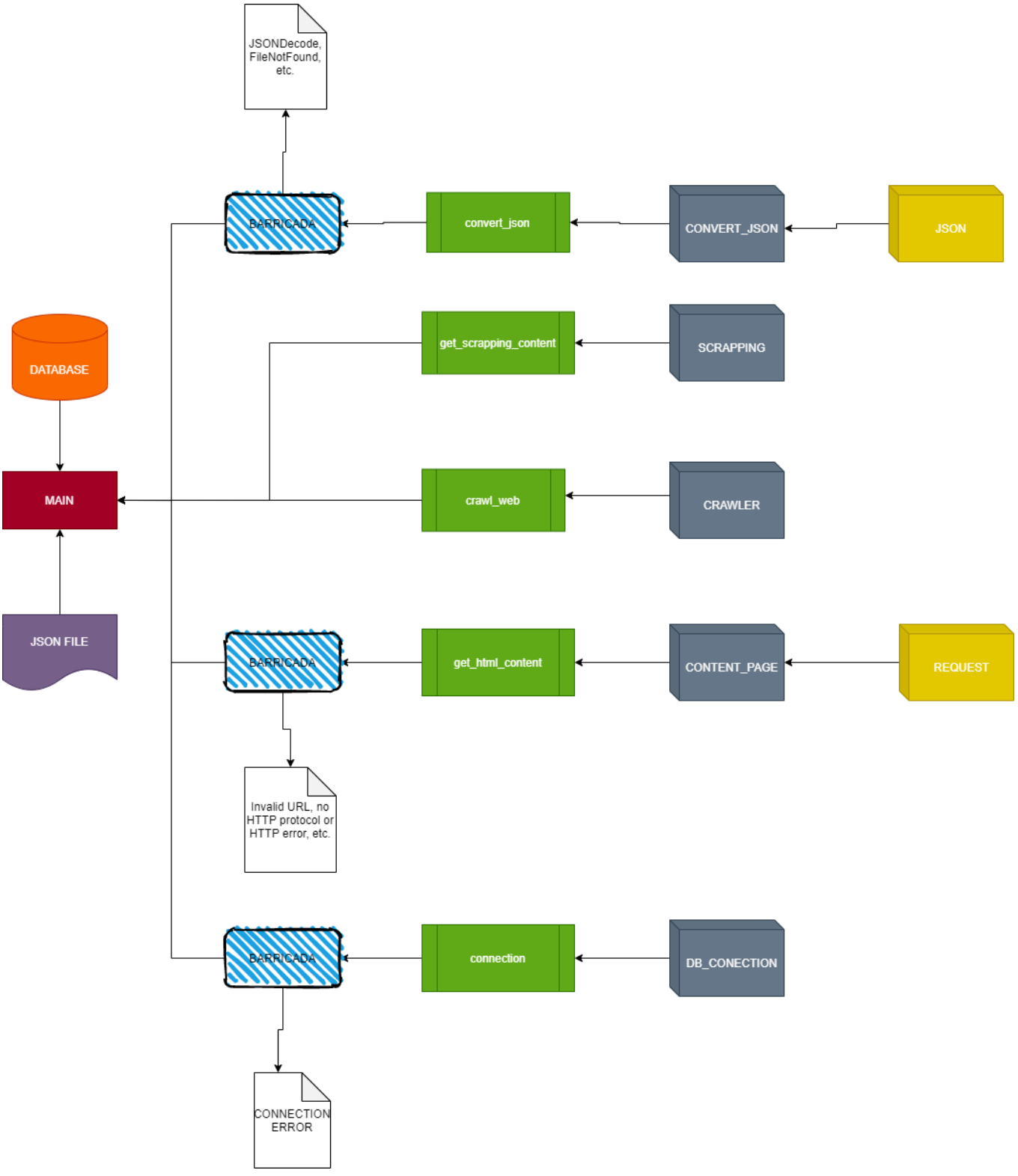
En aquest cas, l'inconvenient major sería el crawler, ja que és un crawler adaptat a la nostra pàgina de manera que no funciona correctament en el cas de voler crawlear un altre pàgina.

Obtenció e instal·lació:

Per a executar el programa s'ha de tenir descarregat python y després procedir a l'execució de l'arxiu main.py que es pot executar per la terminal de Visual Studio Code.

Arquitectura del sistema

La arquitectura del sistema define no solo la estructura de la aplicación sino como está se verá conectada entre sí para ejecutar el programa. En nuestro proyecto decidimos que lo más óptimo es usar una arquitectura modular ya que nos permite tener una mejor distribución del proyecto en base a modulos/componentes los cuales se llaman a main.py para ejecutarse el programa. En nuestro caso los módulos de este proyecto tienen poco acoplamiento ya que cada módulo es independiente del otro, todos se importan directamente al main.py para simplemente llamarlos y que hagan su tarea y función. Otro punto a tener en cuenta, es que las barricadas que pueden evitar que el programa no siga su ejecución están en tres módulos: database, convert_json y content_page. Debido principalmente a que usan librerías como: json, pymongo y requests que pueden fallar según el error que se presente.


JSONDecode, FileNotFound, etc.
BARRICADA
convert_json
CONVERT_JSON
JSON
DATABASE
get_scrapping_content
SCRAPPING
MAIN
crawl_web
CRAWLER
JSON FILE
BARRICADA
get_html_content
CONTENT_PAGE
REQUEST
Invalid URL, no HTTP protocol or HTTP error, etc.
BARRICADA
connection
DB_CONECTION
CONNECTION ERROR

Funcionalitat de cada mòdul:

crawling.py

- **Propòsit:** El seu propòsit y funció és la de extraure tots els links de les diferents pàgines del nostre lloc web, y guardar-los a una llista.
- **Restriccions**: La seva restricció principal es que està orientada cap a la nostra pàgina y que per a poder funcionar per un altre pàgina sería necessari modificar el codi.
- **Dependencies:** No s'ha usat cap dependència externa.
- La seva implementació es troba al main.py

url del repo:
https://github.com/Albertomanas/RicksyProject/blob/master/backend/src/crawler/crawling.py

scrapping.py

- **Propòsit:** El seu propòsit y funció és la de obtenir les dades que vulguem del HTML.
- **Restriccions:** La funció get_scrapping_content la cual s'importa en el main.py només compleix la seva tasca de fer scrapping de les pàgines de la nostra web, es a dir, es un funció únicament creada amb el fí de fer scrapping damunt la nostra pàgina web. Però, hi ha altres funcions les cuals poden ser importades a main.py i que poden ser reutilitzades per a extraure no només dades del contingut de menús si no de la resta de pàgines de la web, aquestes són; get_all_labels, get_content_attribut, find_content.
- **Dependencies:** No s'ha usat cap dependència externa.
- La seva implementació es troba al main.py

url del repo:
https://github.com/Albertomanas/RicksyProject/blob/master/backend/src/scrapping/scrapping.py

convert_json.py

- **Propòsit:** El seu propòsit y funció es la de convertir un diccionario a json y enviar aquestes dades a un arxiu json ja definit.
- **Restriccions:** Compleix amb la seva funció.
- **Dependències:** Es necesaria la llibreria json per a que funcioni, aquesta llibreria es la que ens permet convertir un diccionario a json.
- La seva implementació es troba al main.py

https://github.com/Albertomanas/RicksyProject/blob/master/backend/src/convert_json/convert_json.py

db_connection.py

- **Propòsit:** El seu propòsit es conectarse a una base de dades(mongo) y poder transferir les nostres dades.
- **Restriccions:** Només es podrà connectar amb una base de dades tipo mongo.
- **Dependències:** Es necesaria la llibreria pymongo que és la que permet la connexió entre la base de dades y el nostre programa.
- La seva implementació es troba en el main.py

url del repo:

https://github.com/Albertomanas/RicksyProject/blob/master/backend/src/database/db_conection.py

content_page.py

- **Propòsit:** El seu propòsit es convertir el contingut dels links en un string.
- **Restriccions:** Compleix amb la seva funció.
- **Dependències:** Es necesaria la llibreria requests que permet el correcte funcionament de la conversió del link a string(contingut HTML)
- La seva implementació es troba en el main.py

url del repo:
https://github.com/Albertomanas/RicksyProject/blob/master/backend/src/content_page/content_html.py

main.py

- **Propòsit:** En aquest mòdul es criden a tots el altres mòduls per fer funcionar el programa complet.
- **Restriccions:** Es necessiten els mòduls necesaris per a que funcioni el programa.
- **Dependencias:** Depèn de tots els altres mòduls.

url del repo:
https://github.com/Albertomanas/RicksyProject/blob/master/backend/src/main.py

Planificación y duración.

En este enlace se encuentra la documentación de Clokify:

url del repo:
https://github.com/Albertomanas/RicksyProject/blob/master/docs/Clockify_Summary_Report_11_13_2020-12_10_2020.pdf

Dificultats y posibles millores.

Las dificultats principals han estat la cerca d'informació relacionada al scrapping sense l'ús de cap llibreria, també hem perdut temps a l'hora de pensar casos tests.

A nivell de millores podríem haver fet ús de les expressions regulars però per falta de temps no ha estat possible, també una millor optimització del crawler i afegir elements nous a lo que respecta al front-end.

Bibliografía.

- Multitud de videos a youtube (com...
- https://stackoverflow.com/
- https://www.w3schools.com/
- Arxius de slack, documentació del panel d'activitats...

Conclusió

- Aquest projecte ens ha permès saber com funciona un projecte de desenvolupament, també com usar les eines necessàries per a construir el projecte, a la vegada d'usar metodologies que ens permeten gestionar i organitzar cada part del projecte.
- Em après la importància d'usar casos tests i com la TDD permet no només un òptim desenvolupament del projecte, sinó que ens asegura la funcionalitat del projecte i poder afegir noves funcionalitats sense la preocupació de possibles errors relacionats amb els casos tests.
- Aquest projecte en ha permès veure com totes les assignatures es complementen entre elles amb tot l'aprengut en elles.
- Hem après a com gestionar millor el temps d'un projecte mitjançant clockify i com influeix el temps dedicat a cada part del projecte.

En definitiva hem après molt!!