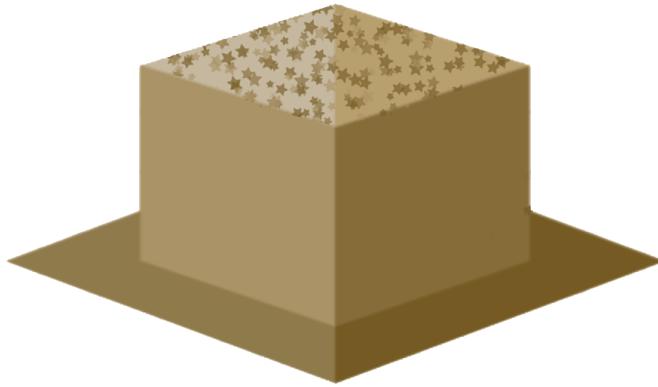


Ian Mercadal

Abril Frongia



PLUMBUS PACKS

| | |
|---|----------|
| Definición del proyecto SW | 3 |
| Requerimientos | 3 |
| Sistemas Informáticos | 3 |
| Bases de datos | 4 |
| Lenguaje de Marcas | 4 |
| Entornos de Desarrollo | 5 |
| Programación | 6 |
| Descripción técnica | 7 |
| Arquitectura de la aplicación y tecnologías utilizadas y diagrama de componentes: | 7 |
| Herramientas y tecnologías utilizadas: | 8 |
| Metodología: | 8 |
| Clockify | 9 |
| Conclusiones | 9 |

Definición del proyecto SW

Nuestro proyecto consiste en un SW que nos mostrará a través de un main.py (que contiene **Crawler** y un **Scraper**), la información y links de nuestra página web, además de guardarla en nuestra base de datos.

Requerimientos

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.

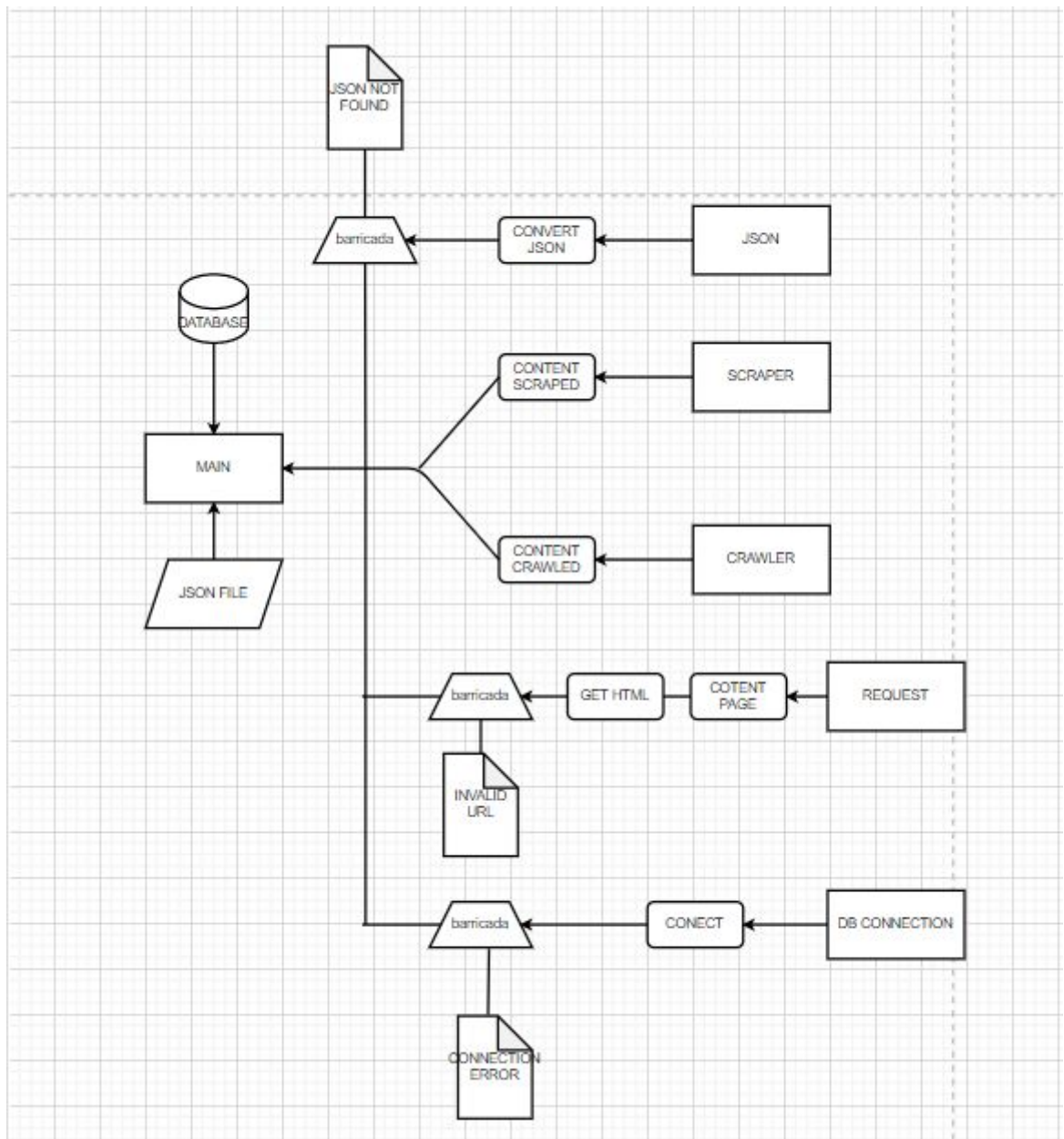
- Divide el código en rutinas de modo que sea SRP y OCP.
 - SRP (S) o Principio de Única Responsabilidad o Single Responsibility Principle. Un componente sólo debe tener un motivo para cambiar.
 - OCP (O) o Open/Closed Principle. Las entidades de software (clases, módulos, funciones, etc.) deben estar “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.

Descripción técnica

A. Arquitectura de la aplicación y tecnologías utilizadas y diagrama de componentes:



Herramientas y tecnologías utilizadas:

Hemos utilizado **Visual Studio Code** para el desarrollo del código, tanto para realizar la página web como el SW (Crawler y Scraper).

Respectivamente, el VS Code tenía un linter de pycode style y una serie de extensiones como **Conventional Commits**, **Git Graph** o **HTML Boilerplate**. De otro modo, para la base de datos hemos usado **Mongo Compass** y el **Atlas**. También hemos hecho uso de Github, tanto para el repositorio como para el dominio de la web como Github Pages.

Tecnologías usadas:

- Visual Studio Code
- HTML5
- CSS3
- Markdown
- Mongo Compass
- Github
- Python

Dependencias:

- Urllib3
- mongopy
- pytest

Metodología:

Hemos hecho la planificación a través del modelo en cascada, que se basa en acabar primero un módulo y este será el resultante del siguiente módulo. El motivo para haber utilizado esta metodología es debido a que para realizar el SW de crawl y scrap, debemos realizar primero una página web a la que aplicar dicho SW. Previamente, para hacer la Web, teníamos que hacer un repositorio donde realizarla y cada uno contribuir con su parte.

Primero de todo, instalamos las herramientas necesarias para la realización del proyecto, respectivamente creamos el repositorio y la base de datos con su respectivo esquema de validación. A partir de ese momento, realizamos la Web y finalmente el SW.

Clockify

A. Diagrama



B. Análisis y duración del tiempo

Hemos dedicado un bastante tiempo a la web y a la parte de programación. Entornos de desarrollo tiene un tiempo algo excesivo por los problemas que nos han generado las branch y documentación. Bases de datos también tiene un tiempo alto debido a que nos dió ciertos problemas el esquema de validación a la hora de introducirlo. En la base de datos, algo parecido pasó con el Mongo Compass.

Conclusiones

A. Posibles mejoras:

- La primera sería la cantidad de tiempo usada para el proyecto. Hemos superado las 80 horas cuando quizás podían haber sido menos, pero nos hemos topado con una serie de problemas y nos han retrasado considerablemente.
- Mejorar el código:
Refactorizar nuestro código de forma que quede más comprimido o más fácil de leer.

c. Uso de ramas.

Al principio realizamos una serie de errores graves y no sabíamos movernos muy bien hasta que nos lo explicaron. Aun así, hemos realizado un mal funcionamiento de estas debido a la inexperiencia.

B. Principales dificultades encontradas:

- Realizar el crawler
- Scraper
- Página Web
- Ramas