# Proyecto Ricksy Bussiness - documentación

# 

# 

# 

# 

**Autores: Víctor Porlan y Berta Vergés**

**Curso: 1o DAW Dual**

**Instituto: CIFPFBMoll**

# ÍNDICE

[**Definición general del proyecto de Software.**](#_kliy1k7pnxls) **3**

[**Especificación de requerimientos del proyecto.**](#_50vwevwwjj93) **3**

[SISTEMAS INFORMÁTICOS.](#_5i140o7cbdl4) 3

[ENTORNOS DE DESARROLLO](#_m8rakcoh188p) 3

[LENGUAJE DE MARCAS](#_lzxspkbom6cg) 4

[PROGRAMACIÓN](#_qrx9mwbkly0a) 4

[BASES DE DATOS](#_ifje2l73prd1) 4

[**Procedimientos de desarrollo.**](#_4g1cscc9cy73) **5**

[HERRAMIENTAS UTILIZADAS](#_tdf11elb62o0) 5

[PLANIFICACIÓN](#_ubvu4tx7zt1q) 5

[**Instalación y prueba.**](#_sk14ptkodq9v) **5**

[REQUISITOS](#_kfexr2npkylu) 5

[OBTENCIÓN E INSTALACIÓN](#_19kvihd5a0er) 6

[**Arquitectura del sistema.**](#_y3crajzcu6n7) **6**

[DIAGRAMAS DE COMPONENTES](#_70i7a8xmrnsa) 7

[**Bibliografía.**](#_e5u3dt7j01t1) **7**

# 

# 

# Definición general del proyecto de Software.

Para el presente proyecto hemos creado un sitio web (usando html y css) de venta de paquetes de bienvenida para una fiesta con temática de Rick y Morty. Una vez hecha la página y añadida toda la información, hemos desarrollado un programa en Python 3 capaz de, a partir de una página raíz, extraer todos los enlaces del sitio web (obviando los que están repetidos) mediante un crawler, y scrapear toda la información de los paquetes de bienvenida (esto es, cuál es su precio, qué objetos contienen, etc.) para meterla en un diccionario y la subirla a una base de datos MongoDB en formato Json. Para conseguir todo esto hemos usado librerías como urllib, request o pymongo.

# Especificación de requerimientos del proyecto.

## SISTEMAS INFORMÁTICOS.

-Evaluar SI, identificar componentes y características.

-Instalar SI(venv, por ejemplo).

-Gestionar información aplicando medidas para la integridad de los datos.

-Gestionar SI mediante comandos y herramientas gráficas.

-Instalar desde CLI todo el software necesario (python, pip, html5, css3, Mongo DB...).

-Invocar desde CMD programa Python y averiguar qué parámetros admite.

-Averiguar qué errores genera una mala invocación.

-Instalar el entorno de desarrollo, extensiones y herramientas de testing y debugging

-Instalar librerías necesarias.

-Instalar desde cmd el control de versiones git.

-Utilizar el VCS desde CLI.

-Crear estructura de directorios desde CMD.

-Copiar o mover desde CMD documentos a la biblioteca.

## ENTORNOS DE DESARROLLO

-Crear un repositorio en github

-Utilizar git, usar las ramas (git flow).

-Crear fichero Readme.md usando la sintaxis markdown.

-Reconocer los elementos y herramientas de desarrollo (git, clockify, conventional commits, IDE) y ponerlos en funcionamiento.

- Evaluar contornos integrados de desarrollo y analizar sus características para editar código y generar archivos ejecutables.

-Verificar funcionamiento de programas y realizar pruebas.

-Realizar el control de versiones para trabajar en pareja.

- Optimizar código utilizando las herramientas de desarrollo necesarias.

-Elaborar la documentación.

-Utilizar correctamente el entorno virtual y requirements.txt.

## LENGUAJE DE MARCAS

-Reconocer características de LM, analizando e interpretando fragmentos de código.

-Utilizar html y css (js opcional) para crear una web estática.

-Utilizar LM para la transmisión de información de la web identificando sus elementos.

-Utilizar las estructuras CSS y html para ayudarnos con el scraping.

- Utilizar el esquema propuesto (acordado con el mentor) para

## PROGRAMACIÓN

-Reconocer la estructura de un programa.

- Escribir y probar programas utilizando programación modular y estructurada.

-Depurar código.

-Controlar errores mediante casos test y/o aserciones a modo de barricada (precondiciones y postcondiciones).

- Realizar operaciones de entrada y salida, utilizar librerías.

-Utilizar estructuras avanzadas de datos para hacer programas que manipulen información.

-Utilizar bloques try/ except para controlar excepciones.

-SRP, OPC y KIS.

-Hacer un crawler.

-Scrapear datos de nuestra web.

-Crear documentación de documentos Json.

-Invocar desde consola el programa.

## BASES DE DATOS

-Reconocer los elementos de las bases de datos.

-Consultar y modificar la info almacenada en una base de datos utilizando asistentes, herramientas gráficas y el lenguaje de manipulación de datos (MongoDB).

-Desarrolla procedimientos almacenados, evaluando y utilizando las sentencias del lenguaje incorporado en el gesto de BBDD.

-Crear un JSON como resultado del scrapping y subirlo a MongoDB.

- Definir el tipo de dato de cada campo del documento.

-Crear una base de datos en la nube mediante el servicio Atlas de MongoDB.

- Definir la autenticación y autorización sobre la BD.

-Definir las colecciones de documentos que consideremos oportunas-

-Utilizar Compass para consultar la info almacenada en la BD y visualizar estadísticas.

# Procedimientos de desarrollo.

## HERRAMIENTAS UTILIZADAS

Para el desarrollo del código hemos utilizado Visual Studio Code, corrigiendo los errores usando un linter de python para detectar los errores humanos durante el desarrollo de la aplicación.

A la hora de hacer los commits a GitHub hemos utilizado Conventional Commits, el cual establece un protocolo adecuado a la hora de realizarlos para facilitar el entendimiento entre los usuarios que desarrollan la aplicación.

## PLANIFICACIÓN

La metodología de desarrollo utilizada ha sido la de cascada, ya que fuimos desarrollando el proyecto en diferentes módulos los cuales debían estar completados para poder pasar al siguiente, ya que cada módulo normalmente dependía del anterior.

Debida a la inexperiencia a la hora de desarrollar una aplicación volvimos sobre nuestros pasos en múltiples ocasiones para refactorizar el código y añadir nuevas funciones para acortar el código, ya que ciertas partes del código que se repetían se podrían haber puesto en una función y haberla invocado cada vez que se requiriera su uso.

Los primeros días del proyecto los dedicamos a planear la estructura de la web y a aclarar las posibles características que tendrían los paquetes dependiendo de su calidad (Basic, Standard y Premium). Una vez aclarada la estructura decidimos cual nos parecía la forma más lógica de desarrollar un scraper con python, basándonos en gran parte a las funcionalidades de Beautifulsoup y la guía de gestión de datos de Udacity.

# Instalación y prueba.

## REQUISITOS

Es necesario instalar las librerías listadas en el documento requirements.txt para poder usar adecuadamente la aplicación. Para el uso de los casos test dentro de la carpeta pytest es necesaria la instalación de la librería Pytest, la cual no ha sido añadida en requirements.txt ya que no es vital para hacer uso de la aplicación.

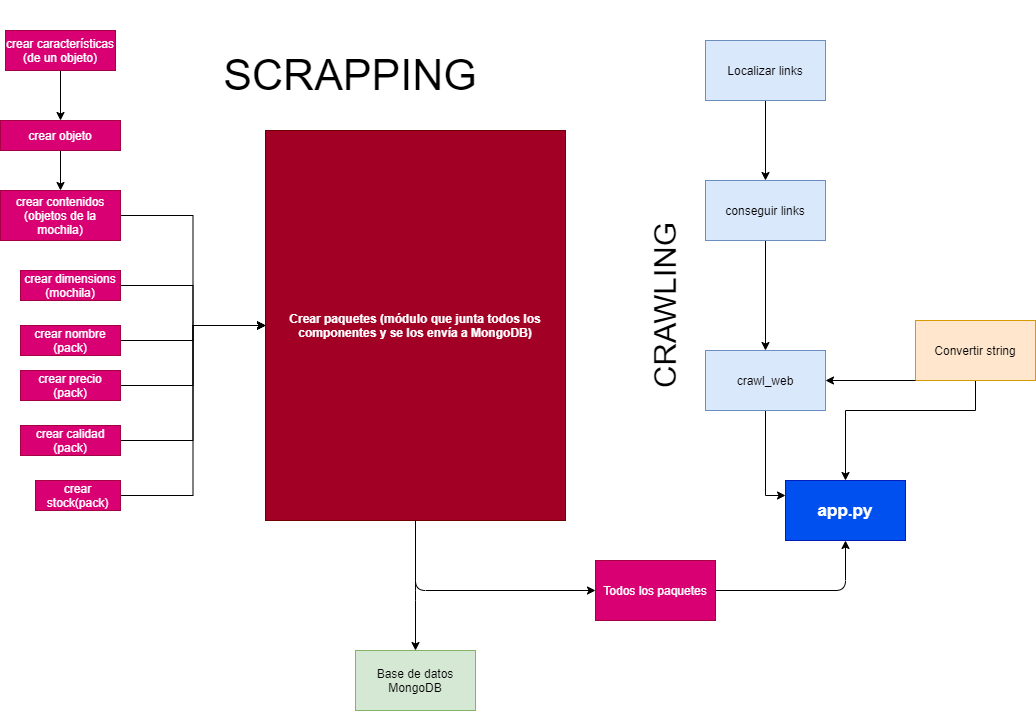
## OBTENCIÓN E INSTALACIÓN

El programa se encuentra dentro del directorio services, y el archivo a ejecutar es app.py. Para su ejecución simplemente debemos ejecutarlo desde una terminal, (Ya sea la de Visual Studio Code o la de python) escribiendo la ruta y el nombre del archivo (instrucciones más detalladas incluidas en el README).

# Arquitectura del sistema.

Hemos creado una estructura modular prácticamente basada en microservicios, excepto alguna función (como la de crear paquetes, la cual integra todos los componentes de los paquetes y se lo envía a MongoDB) que depende de varias otras.

## DIAGRAMAS DE COMPONENTES



Convertir string es la función que usa el URLopen para sacar el html de una página.

Los módulos en color azul bebé son el crawling . El primero (localizar\_links) encuentra la posición de los links (devuelve url y dónde acaba ésta), conseguir\_links añade los links a una lista, y crawl\_web repite todo esto para que haga lo mismo en todas las páginas del sitio web.

Los módulos de color rosa son el scrapping: añaden ítems al diccionario (ya sea calidad, precio, nombre…).

El módulo en rojo integra todos los de color rosa y sube los packs a mongoDB (uno a uno, con insert one). Todos\_los\_paquetes repite crear\_paquetes recoge todos los paquetes de cada link.

app.py le pasa a la función todos\_los\_paquetes los 4 links recogidos anteriormente por crawl\_web para que lo repita una vez por cada link encontrado.

# Bibliografía.

-Documentación oficial de las herramientas de desarrollo y lenguajes.

- Stack overflow.

-Slack, drive(Udacity, cursos de debugging, cursos de ramas, git flow, libro de la rata...) y ayuda de profesores, compañeros y el mentor.

-Youtube.

- Al Sweigart - Automate The Boring Stuff With Python\_Practical Programming For Total Beginners-No Starch Press (2019), segunda edición