

UNIVERSIDAD DEL BÍO-BÍO
Facultad de Ciencias Empresariales
Departamento de Sistemas de Informaciones.



UNIVERSIDAD DEL BÍO-BÍO

Cliente web para la extracción de contenido de diversas páginas web utilizando los servicios incorporados en la *API TECO*.

MEMORIA PARA OPTAR A TÍTULO DE INGENIERO DE EJECUCIÓN EN
COMPUTACIÓN E INFORMÁTICA.

Alumno: Jorge Vallejos Prado.

Profesor Guía: Alejandra Segura Navarrete.

Concepción, 2019.

Índice General

1.	Introducción	9
1.1.	Definiciones, siglas y abreviaciones	11
2.	Definición del proyecto	12
2.1.	Definición de la institución	12
2.2.	Origen del tema	14
2.3.	Objetivos del proyecto	14
2.3.1.	Objetivo general	14
2.3.2.	Objetivos específicos	14
2.4.	Aportes	15
2.5.	Alcances y Límites.....	15
3.	<i>API TECO</i>	16
3.1.	Funcionalidad de los Servicios de la API	16
3.2.	Encabezado de Peticiones	17
3.3.	Servicios Cuentas de Usuarios	17
3.3.1.	Servicio de Registro.....	17
3.3.2.	Servicio de Activación.....	18
3.3.3.	Servicio de Ingreso	18
3.3.4.	Servicio de Petición reinicio de contraseña	18
3.3.5.	Servicio de Reinicio de contraseña.....	19
3.3.6.	Servicio de Reinicio de correo no válido	19
3.4.	Servicios Usuario	19
3.4.1.	Servicio de Información.....	19
3.4.2.	Servicio de Extractores	19
3.4.3.	Servicio Para Agregar extractor	19
3.4.4.	Servicio Para Modificar extractor.....	20
3.4.5.	Servicio Para Eliminar extractor.....	21
3.5.	Servicios Extractores	21
3.5.1.	Servicio de Extractores Compartidos	21
3.5.2.	Servicio Para Buscar por ID.....	21
3.5.3.	Servicio Para Buscar por Nombre	21
3.5.4.	Servicio Para Buscar por Página Web	21
3.5.5.	Servicio Para Ejecutar extractor	21
3.5.6.	Servicio Para Probar extractor.....	22
3.5.7.	Servicio Para Ejecutar Nodo	22
3.5.8.	Servicio Para Probar Nodo.....	23
3.5.9.	Servicio de Extractor de Twitter.....	23

3.6.	Servicios Extracciones Programadas	24
3.6.1.	Servicio de Extracciones Programadas Personales	24
3.6.2.	Servicio Para Buscar por ID.....	24
3.6.3.	Servicio Para Eliminar programación.....	24
3.6.4.	Servicio de Extractor diario.....	24
3.6.5.	Servicio de Extractor semanal.....	25
3.6.6.	Servicio de Extractor mensual.....	25
3.6.7.	Servicio de Nodo diario	25
3.6.8.	Servicio de Nodo semanal.....	26
3.6.9.	Servicio de Nodo mensual	26
4.	Sistema: Integración de los servicios incorporados en la API TECO en una aplicación web.27	
4.1.	Ambiente de Ingeniería de Software	27
4.1.1.	Metodología de Desarrollo.....	27
4.1.2.	Estándares de documentación	27
4.1.3.	Técnicas y notaciones	27
4.1.4.	Herramientas de apoyo al desarrollo de software que serán utilizadas	28
4.1.5.	Hardware para el desarrollo de software.....	28
4.2.	Especificación de Requerimientos.....	29
4.2.1.	Objetivo General del Software	29
4.2.2.	Objetivos Específicos del Software	29
4.2.3.	Descripción global del producto.....	29
4.2.4.	Requerimientos funcionales del sistema.....	30
4.2.5.	Requerimientos no funcionales	31
4.2.6.	Interfaces externas de entrada	31
4.2.7.	Interfaces externas de salida.....	32
4.3.	Estudio de Factibilidad.....	32
4.3.1.	Factibilidad técnica.....	33
4.3.2.	Factibilidad Económica.....	34
4.3.3.	Factibilidad Operativa.....	34
4.3.4.	Conclusión Factibilidad	34
4.4.	Análisis.....	34
4.4.1.	Diagramas de casos de Uso.....	34
4.4.2.	Modelamiento de Datos	44
4.5.	Diseño y construcción.....	45
4.5.1.	Diseño de arquitectura funcional.....	45
4.5.2.	Diseño interfaz y navegación	46
5.	Pruebas	53

5.1.	Elementos de prueba	53
5.2.	Especificación de pruebas	53
5.3.	Responsable de las pruebas	53
5.4.	Detalle de las pruebas	54
5.5.	Conclusiones de las pruebas.....	58
6.	Conclusiones	59
7.	Bibliografía	60
8.	Anexo: Estimación por puntos de casos de uso.....	61
8.1.	Calcular UAW (unadjusted actor weights)	61
8.2.	Calcular UUCW (unadjusted use case weights).....	61
8.3.	Calcular TCF (technical complexity factor).....	62
8.4.	Calcular ECF (environmental complexity factor)	63
8.5.	Calcular UCP (use case points)	63
8.6.	Estimación final	64
9.	Anexo: Recursos utilizados API teco.....	65
9.1.	Recursos requeridos	65
9.2.	Objetos requeridos	69

Índice Tablas

Tabla 1 Hardware utilizado para el desarrollo	28
Tabla 2 Requerimientos funcionales del sistema.....	30
Tabla 3 Requerimientos no funcionales.....	31
Tabla 4 Interfaces externas de entrada.....	31
Tabla 5 Interfaces externas de salida	32
Tabla 6 Herramientas de Desarrollo	33
Tabla 7 Caso de uso: Iniciar Sesión.....	36
Tabla 8 Caso de Uso: Crear Cuenta.....	37
Tabla 9 Caso de uso: Activar Cuenta.....	38
Tabla 10 Caso de Uso: Crear Extractor.....	39
Tabla 11 Caso de Uso: Ejecutar Extractor	40
Tabla 12 Caso de Uso: Programar Extractor.....	41
Tabla 13 Caso de Uso: Extraer desde Twitter.....	42
Tabla 14 Caso de Uso: Generar CSV de Textos	43
Tabla 15 Trazabilidad de casos de uso.....	43
Tabla 16 Prueba Extraer desde Twitter.....	54
Tabla 17 Prueba Generar Extractor (a)	56
Tabla 18 Prueba Generar Extractor (b).....	57
Tabla 19 Cálculo de pesos de actores sin ajuste.....	61
Tabla 20 Cálculo de pesos de casos de uso sin ajuste.....	61
Tabla 21 Cálculo de factor de complejidad técnico.....	62
Tabla 22 Cálculo de factor de complejidad ambiental.....	63
Tabla 23 Estimación de esfuerzo total.....	64
Tabla 24 Servicio Cuentas.....	65
Tabla 25 Servicio Usuarios	66
Tabla 26 Servicio extractores	68
Tabla 27 Servicio Programar Extractores	68
Tabla 28 Objeto Nuevo Usuario	69
Tabla 29 Objeto Reiniciar Contraseña	69

Tabla 30 Objeto Cambiar Email	70
Tabla 31 Objeto Datos Extractor	70
Tabla 32 Objeto Actualizar Datos Extractor	70
Tabla 33 Objeto Nodo.....	71
Tabla 34 Objeto Elemento.....	71
Tabla 35 Objeto Probar Datos Extractor.....	71
Tabla 36 Objeto Probar Nodo.....	72

Índice Figuras

Ilustración 1 Estructura Organizacional.....	13
Ilustración 2 Ejemplo petición JSON para registro	17
Ilustración 3 Ejemplo petición JSON para registro	18
Ilustración 4 Ejemplo petición JSON para petición de reinicio de contraseña	18
Ilustración 5 Ejemplo petición JSON para reinicio de contraseña	19
Ilustración 6 Ejemplo petición JSON para reinicio de correo no valido	19
Ilustración 7 Ejemplo petición JSON para crear un extractor.....	20
Ilustración 8 Ejemplo petición JSON para modificar un extractor.....	20
Ilustración 9 Ejemplo petición JSON para ejecutar un extractor	21
Ilustración 10 Ejemplo petición JSON para probar un extractor.....	22
Ilustración 11 Ejemplo petición JSON para ejecutar un nodo	22
Ilustración 12 Ejemplo petición JSON para probar un nodo.....	23
Ilustración 13 Ejemplo peticiones JSON para extraer contenido de Tweets	23
Ilustración 14 Ejemplo petición JSON para eliminar una programación	24
Ilustración 15 Ejemplo petición JSON para programar una extracción diaria.....	24
Ilustración 16 Ejemplo petición JSON para programar una extracción semanal	25
Ilustración 17 Ejemplo petición JSON para programar una extracción mensual.....	25
Ilustración 18 Ejemplo petición JSON para programar una extracción diaria nodo	25
Ilustración 19 Ejemplo petición JSON para programar una extracción semanal nodo	26
Ilustración 20 Ejemplo petición JSON para programar una extracción mensual nodo	26
Ilustración 21 Caso de Uso.....	35
Ilustración 22 Representación JSON de la API.....	44
Ilustración 23 Adaptación JSON a Entidad-Relación.....	45
Ilustración 24 Vista ingreso al sistema	46
Ilustración 25 Vista mis extractores.....	47
Ilustración 26 Vista extractores públicos	48
Ilustración 27 Vista resultados de ejecución	48
Ilustración 28 Vista búsqueda en Twitter.....	49
Ilustración 29 Vista nuevo extractor.....	50

Ilustración 30 Vista resultado extracción programada..... 51

1. INTRODUCCIÓN

Una de las líneas de las cuales trabaja el grupo de investigación SOMOS es el “Análisis de subjetividad en textos” centrada en analizar opiniones expresadas a través de textos extraídas desde redes sociales u otras fuentes de textos de los cuales se obtienen los corpus sobre los cuales se realizan los análisis de la minería de opinión, el análisis afectivo, detección de agresividad, entre otros. Considerando lo anterior se desarrolló la API TECO, la cual permite la extracción de texto desde Twitter y extracciones de diferentes páginas webs para las cuales cada usuario puede crear extractores. Ya que la API solamente pone a disponibilidad de los usuarios los servicios para realizar las extracciones esto se vuelve una labor tediosa para un usuario común pues, para empezar, como algo básico se necesita configurar el encabezado de las peticiones, cosa que no cualquier persona sabrá cómo realizar. Debido a este panorama nace la idea de desarrollar una aplicación web que sea capaz de reunir los servicios que se encuentran disponible en la API y presentarlos a los usuarios de una forma más agradable y sencilla de utilizar, dejando de forma interna todo lo que conlleva la realización de peticiones http hacia la API. Junto con esto se espera que la aplicación web pueda tener una buena aceptación y masificar su uso.

En este documento se detallará el desarrollo del proyecto, el cual estará dividido en varios capítulos siendo tres de ellos los principales que se especifican a continuación:

En el segundo capítulo se realiza la definición del proyecto, en el cual se especificarán puntos como el origen del tema, los objetivos del proyecto, sus aportes, alcances y límites.

El tercer capítulo se describe la API TECO, las funcionalidades y servicios que esta posee y la función que cumplen cada uno de ellos.

Como cuarto capítulo se describe el desarrollo del sistema que incorporara los servicios de la API TECO, en el contenido de este capítulo, podremos encontrar detalles del ambiente de ingeniería de software, la especificación del requerimientos que contempla los objetivos del software y definición de requerimientos, además de secciones dedicadas al estudio de

9

factibilidad, análisis, donde se lleva a cabo un adaptación de los requerimientos a casos de uso y modelamiento de datos y como último lugar el diseño, en donde se describe la arquitectura e interfaz del sistema.

En el quinto capítulo se detallarán las pruebas unitarias realizadas al sistema.

1.1. DEFINICIONES, SIGLAS Y ABREVIACIONES

API: Application Programming Interface (Interfaz de Programación de Aplicaciones).

CSV: Comma Separated Value (valores separados por comas).

Formato JSON: JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript.

Patrón Adapter: Patrones estructural de diseño de software, permite establecer una capa intermedia que permita comunicarse a dos interfaces que de otro modo no podrían hacerlo, realizando una adaptación de la interfaz que proporciona el servicio a la que la solicita.

Framework Laravel: Es un framework de código abierto que permite el desarrollo de aplicaciones y servicios web con PHP 5 y PHP 7.

HTTP: Protocolo de transferencia de hipertexto, define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse.

SOMOS: Software Modelling & Science

Front-End: Parte del software que interactúa con los usuarios.

Back-End: Parte del software que se conecta con la base de datos y el servidor que utiliza un sitio web.

Token: Autorización que permite el acceso a recursos protegidos.

2. DEFINICIÓN DEL PROYECTO

El grupo de investigación SOMOS cuenta con una API que permite la extracción de texto de diferentes páginas webs, la intención del proyecto es la implementación de una interfaz gráfica que permita el uso de los recursos de extracción desde un solo lugar de forma simplificada y poder la aplicación web a disponibilidad de un público objetivo y poder hacer uso de esta a modo de aplicación y no solamente como una interfaz.

2.1. DEFINICIÓN DE LA INSTITUCIÓN

Misión.

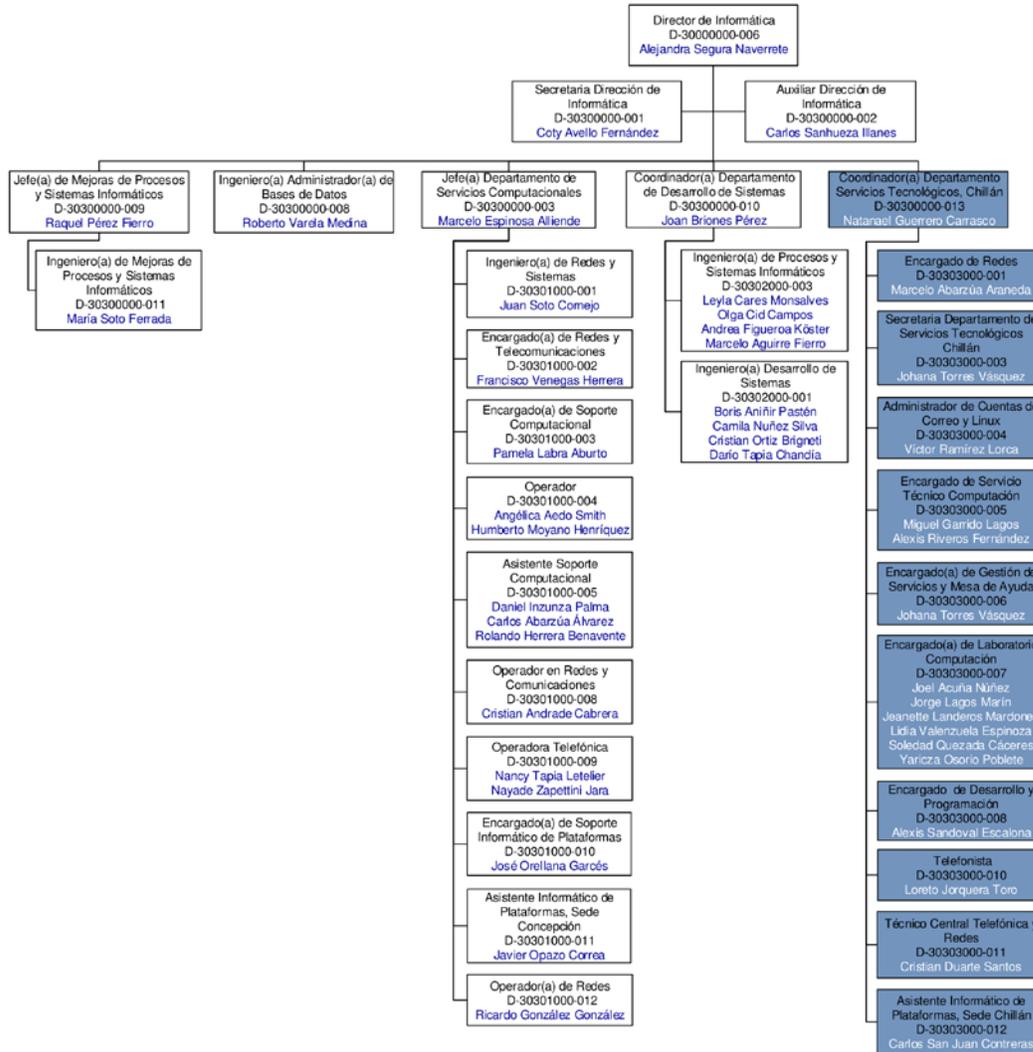
La Universidad del Bío-Bío, a partir de su naturaleza pública, responsable socialmente y estatal, tiene por misión, desde las Regiones del Biobío y Ñuble, aportar a la sociedad con la formación de personas integrales, a través de una Educación Superior de excelencia. Comprometida con los desafíos regionales y nacionales, contribuye a la movilidad e integración social por medio de; la generación y transferencia de conocimiento avanzado, mediante la docencia de pregrado y postgrado de calidad, la investigación fundamental, aplicada y de desarrollo, la vinculación bidireccional con el medio, la formación continua y la extensión. Asimismo, impulsa el emprendimiento y la innovación, el fortalecimiento de la internacionalización y el desarrollo sustentable de sus actividades, basada en una cultura participativa centrada en el respeto a las personas.

Visión.

Ser reconocida a nivel nacional e internacional como una Universidad pública, responsable socialmente y regional que, comprometida con su rol estatal, desde las Regiones del Biobío y Ñuble, forma personas integrales de excelencia y aporta a través de su quehacer al desarrollo sustentable de las regiones y el país.



Organigrama de la Dirección de Informática



Sistema de Gestión de la Calidad – Departamento de Normalización y Certificación

Fuente: Universidad del Bío-Bío, Organigrama de la Dirección de Informática (2018)

Ilustración 1 Estructura Organizacional

2.2. ORIGEN DEL TEMA

El grupo de investigación SOMOS, trabajan en 5 líneas investigativas, una de ella es “Análisis de subjetividad en textos” la cual se aplica al análisis de opiniones texto expresadas en diversas fuentes de las redes sociales. Considerando lo anterior se implementó la API TECO que permite extraer comentarios textuales desde Twitter o una página web, generando los corpus sobre los cuales se realizan los análisis de la minería de opinión, el análisis afectivo, detección de agresividad, etc.

La API TECO cuenta principalmente con servicios de extracción de contenido desde Twitter y también desde alguna página web en específico, caso en el cual se debe especificar la sección de la página desde la cual se desea extraer el contenido.

Actualmente los servicios de esta API están desplegados en un server docker que permite el manejo de contenedores, y los servicios que ofrece pueden ser accedidos a través de URLs, por lo cual su uso depende de aplicaciones de terceros que puedan realizar peticiones HTTP con herramientas como Postman o Insomnia, entre otras, que permiten probarlos. Lo anterior conlleva un proceso muy tedioso que debe ser desarrollado por un informático desde el registro de usuario hasta la ejecución de los extractores. Aún no existen aplicaciones web que consuman dichos servicios.

2.3. OBJETIVOS DEL PROYECTO

2.3.1. Objetivo general

Desarrollar una aplicación web que permita disponibilizar los servicios contenidos en la *API TECO* de forma gráfica y sencilla, que permita difundir y masificar su uso.

2.3.2. Objetivos específicos

- Análisis del funcionamiento de los recursos disponibles en la API TECO.
- Desarrollar interfaz gráfica intuitiva que represente las funciones de la API TECO.
- Desarrollar interfaz para que cualquiera pueda crear un extractor sobre un sitio específico.
- Establecer intermediario de comunicación desde la aplicación con la API.
- Generar formato de descarga de contenido.

2.4. APORTES

Desarrollo e implementación front-end de una aplicación web que facilite el uso de extractores de texto, tomando como base los servicios disponibles de API TECO, del grupo de investigación SOMOS.

2.5. ALCANCES Y LÍMITES

El proyecto se centra en la integración de los servicios de API TECO dentro de una aplicación la cual permita hacer uso de sus servicios de una manera más simple para los usuarios, sin la necesidad recurrir a otras aplicaciones en las cuales deban saber cómo realizar peticiones http para poder hacer uso de los servicios de extracción con los que cuenta la API TECO.

3. API TECO

La API TECO (*Text Extraction Community*) es un conjunto de funciones desarrollada con la finalidad de disponibilizar servicios para la extracción de contenido desde Twitter y también para que personas puedan crear y ejecutar extractores personalizados de texto de diferentes sitios webs, caso en el cual se especifica la o las secciones de la página de la cual se obtendrá en contenido, todo esto a través de él envío y respuesta de información a través, de JSON en peticiones https para permitir la comunicación con la API.

3.1. FUNCIONALIDAD DE LOS SERVICIOS DE LA API

La API TECO nos permite acceder a diferentes servicios, los cuales están relacionados en su mayoría con la creación, ejecución, y programación de extractores de texto, ya que además de contar con esto también se cuenta con servicios enfocado a la creación y autenticación de usuario, puesto que se necesita contar con un usuario registrado para poder tener acceso a los demás servicios de extracción disponibles. La API solo envía y recibe datos a través de la notación JSON.

Entre sus funciones se pueden encontrar las siguientes:

- Manejo de cuentas de usuario
- Obtención de contenidos desde Twitter mediante consulta ingresada por el usuario
- Creación y ejecución de extractores
- Programación de ejecución para extractores
- Ejecución de extractores público

3.2. ENCABEZADO DE PETICIONES

La API es capaz de interactuar con diferentes aplicaciones que hagan uso sus servicios a través de la notación JSON, es por este motivo que se debe especificar los parámetros del encabezado en cada petición. Cada usuario tiene un token de autenticación, el cual debe ser especificando en el encabezado para utilizar los servicios que requieren de la autorización,

Ejemplo configuración de encabezado:

```
{  
  "Content-Type" : "application/json" ,  
  "x-access-token" : "token de autorización del usuario"  
}
```

3.3. SERVICIOS CUENTAS DE USUARIOS

La sección de cuentas de usuario cuenta con los servicios básicos para la interacción de un usuario con los demás servicios que dispone la API, como son el registro de usuarios, activación de cuenta, ingreso, recuperar cuenta en caso de olvidar su contraseña y cambio de contraseña o correo registrado en la cuenta.

3.3.1. Servicio de Registro

Creación de credenciales para que los usuarios puedan utilizar los servicios que la API tiene disponible.

```
{  
  "login": "usuario",  
  "email": "usuario@gmail.com",  
  "password": "usuario123",  
  "passwordRepeat": "usuario123"  
}
```

Ilustración 2 Ejemplo petición JSON para registro

3.3.2. Servicio de Activación

Verifica y activa las credenciales de un usuario gracias a un código de activación envía al email con el que el usuario se registró.

```
{
  "login": "usuario",
  "email": "usuario@gmail.com",
  "token": "usuariotoken"
}
```

Ilustración 3 Ejemplo petición JSON para registro

3.3.3. Servicio de Ingreso

Valida las credenciales del usuario para entregar un token de autenticación para poder utilizar los servicios de la API.

```
{
  "login": "usuario",
  "password": "usuario123"
}
```

Ilustración 3 Ejemplo petición JSON para ingreso

3.3.4. Servicio de Petición reinicio de contraseña

Si un usuario olvida su contraseña debe ingresar el email con el que realizó su registro, y le será enviado un código para reiniciar su contraseña.

```
{
  "email": "supertester@gmail.com"
}
```

Ilustración 4 Ejemplo petición JSON para petición de reinicio de contraseña

3.3.5. Servicio de Reinicio de contraseña

El usuario debe ingresar su nombre de usuario, email, nueva contraseña y el código enviado a su email para poder registrar su nueva contraseña.

```
{
  "login": "usuario",
  "email": "usuario@gmail.com",
  "newPassword": "usuario123",
  "newPasswordRepeat": "usuario123",
  "token": "usuariotoken"
}
```

Ilustración 5 Ejemplo petición JSON para reinicio de contraseña

3.3.6. Servicio de Reinicio de correo no válido

Permite al usuario reemplazar el email con el que se registró por un email nuevo.

```
{
  "login": "usuario",
  "email": "usuario@gmail.com",
  "password": "usuario123",
  "newEmail": "usuario123@gmail.com"
}
```

Ilustración 6 Ejemplo petición JSON para reinicio de correo no valido

3.4. SERVICIOS USUARIO

Cuenta con los servicios relacionados a los datos de un usuario, permite visualizar la información pública de algún usuario y acceder a los extractores que el usuario haya creado. Además, nos permite agregar, modificar y eliminar extractores personales.

3.4.1. Servicio de Información

Mediante el nombre de un usuario, permite obtener la información pública de este la cual sería su email, nombre de usuario, fecha de registro e identificador.

3.4.2. Servicio de Extractores

Permite recuperar los extractores propios del usuario.

3.4.3. Servicio Para Agregar extractor

Permite al usuario crear un nuevo extractor de texto, especificando exactamente la sección de la página en donde desea obtener el contenido.

```

{
  "extractorData": {
    "name": "Mi extractor",
    "url": "http://www.mixtractor.cl",
    "description": "Es un extractor con nodos.",
    "shared": true,
    "nodes": [
      {
        "name": "Nodo",
        "description": "Es un nodo con elementos.",
        "container": ".container",
        "elements": [
          {
            "tag": "Elemento",
            "selector": ".selector"
          }
        ]
      }
    ]
  }
}

```

Ilustración 7 Ejemplo petición JSON para crear un extractor

3.4.4. Servicio Para Modificar extractor

El usuario puede realizar cambios en un extractor ya creado, haciendo alguna modificación simple como modificar el nombre del extractor, o realizar un cambio más complejo como añadir un nuevo nodo o añadir un nuevo elemento a un nodo ya existente.

```

{
  "extractorDataUpdate": {
    "id": "id del Extractor",
    "nodes": [
      {
        "name": "Nodo",
        "description": "Es un nodo con elementos.",
        "container": ".container",
        "elements": [
          {
            "tag": "Elemento",
            "selector": ".selector"
          },
          {
            "tag": "Elemento 2",
            "selector": ".selector2"
          }
        ]
      }
    ]
  }
}

```

Ilustración 8 Ejemplo petición JSON para modificar un extractor

3.4.5. Servicio Para Eliminar extractor

Permite eliminar un extractor mediante su ID.

3.5. SERVICIOS EXTRACTORES

Los servicios de extractores principalmente nos permiten obtener todos los extractores que los usuarios hayan creado en modo público y ejecutarlos para obtener el contenido de las páginas para las cuales fueron creados. También nos permite probar extractores o nodos antes de ser creados para verificar si su estructura esta correcta o debe ser corregida. Por otra parte, cuenta con función para búsqueda de extractores por ID, Nombre o Página Web sobre la cual extraen contenido.

De igual forma se encuentra el servicio de extracción desde Twitter que permite obtener los Tweets relacionados con la busca realizada.

3.5.1. Servicio de Extractores Compartidos

Permite obtener todos extractores registrados, siempre que estos hayan sido configurados como extractores públicos.

3.5.2. Servicio Para Buscar por ID

Permite recuperar la información de un extractor público mediante el uso de su ID

3.5.3. Servicio Para Buscar por Nombre

Permite recuperar la información de un extractor público mediante el uso de su Nombre

3.5.4. Servicio Para Buscar por Página Web

Permite recuperar la información de un extractor público mediante el uso de la URL para la cual fue creado.

3.5.5. Servicio Para Ejecutar extractor

Inicia el proceso de extracción de contenido en base a un extractor, ya sea público, propio o privado, mediante la especificación del ID del extractor que se desea ejecutar.

```
{
  "extractorID" : "superextractorID"
}
```

Ilustración 9 Ejemplo petición JSON para ejecutar un extractor

3.5.6. Servicio Para Probar extractor

Inicia un proceso de extracción a partir de un extractor ingresado, el cual no se encuentra registrado, para corroborar su funcionamiento.

```
{
  "testExtractorData": {
    "name": "Extractor",
    "url": "http://www.extractor.cl",
    "nodes": [
      {
        "name": "Nodo",
        "container": ".container",
        "elements": [
          {
            "tag": "Elemento",
            "selector": ".selector"
          }
        ]
      }
    ]
  }
}
```

Ilustración 10 Ejemplo petición JSON para probar un extractor

3.5.7. Servicio Para Ejecutar Nodo

Inicia el proceso de extracción sobre un nodo específico de un extractor.

```
{
  "extractorID" : "superextractorID" ,
  "nodeName" : "Super Nodo"
}
```

Ilustración 11 Ejemplo petición JSON para ejecutar un nodo

3.5.8. Servicio Para Probar Nodo

Inicia un proceso de extracción a partir de un nodo ingresado, el cual no se encuentra registrado, para corroborar su funcionamiento.

```
{
  "testNodeData": {
    "url": "http://www.superextractor.cl",
    "node": {
      "name": "Super Nodo",
      "container": ".supercontainer",
      "elements": [
        {
          "tag": "Super Elemento",
          "selector": ".superselector"
        }
      ]
    }
  }
}
```

Ilustración 12 Ejemplo petición JSON para probar un nodo

3.5.9. Servicio de Extractor de Twitter

Inicia un proceso de extracción a partir del buscador Twitter Search, permitiendo obtener los Tweets relacionados desde una búsqueda simple o avanzada a través de los Operadores que ofrece el buscador.

```
Búsqueda simple ( relacionados )
{
  "querySearch" : "chile"
}

Búsqueda avanzada ( referencia )
{
  "querySearch" : "@ubbChile"
}

Búsqueda avanzada ( desde )
{
  "querySearch" : "from:@ubbChile"
}
```

Ilustración 13 Ejemplo peticiones JSON para extraer contenido de Tweets

3.6. SERVICIOS EXTRACCIONES PROGRAMADAS

Permiten programar la ejecución de los extractores o nodos, ya sea para que estos sean ejecutados de forma diaria, semanal o mensual, adicionalmente implementa un buscador de programaciones por ID y también un servicio para eliminar programaciones

3.6.1. Servicio de Extracciones Programadas Personales

Permite buscar todas las programaciones personales que tiene un usuario.

3.6.2. Servicio Para Buscar por ID

Permite buscar una programación específica de un usuario mediante el uso de la ID de la programación.

3.6.3. Servicio Para Eliminar programación

Comprueba la existencia de la programación, para luego eliminar los registros de la colección de extracciones programadas personales del usuario.

```
{
  "scheduleID": "programacionID"
}
```

Ilustración 14 Ejemplo petición JSON para eliminar una programación

3.6.4. Servicio de Extractor diario

Comprueba la existencia del extractor el cual se quiere programar, luego se crea la programación del extractor todos los días a una hora especificada por el usuario.

```
{
  "extractorID": "extractorID",
  "atHour": 23,
  "atMinute": 0
}
```

Ilustración 15 Ejemplo petición JSON para programar una extracción diaria

3.6.5. Servicio de Extractor semanal

Comprueba la existencia del extractor el cual se quiere programar, luego se crea la programación del extractor un día de la semana a una hora especificada por el usuario.

```
{
  "extractorID": "extractorID",
  "dayOfWeek": "Martes",
  "atHour": 23,
  "atMinute": 0
}
```

Ilustración 16 Ejemplo petición JSON para programar una extracción semanal

3.6.6. Servicio de Extractor mensual

Comprueba la existencia del extractor el cual se quiere programar, luego se crea la programación del extractor un día X del mes a una hora especificada por el usuario.

```
{
  "extractorID": "extractorID",
  "dayOfMonth": 15,
  "atHour": 23,
  "atMinute": 0
}
```

Ilustración 17 Ejemplo petición JSON para programar una extracción mensual

3.6.7. Servicio de Nodo diario

Comprueba la existencia del nodo el cual se quiere programar, luego se crea la programación del extractor todos los días a una hora especificada por el usuario.

```
{
  "extractorID": "extractorID",
  "nodeName": "Nodo",
  "atHour": 7,
  "atMinute": 0
}
```

Ilustración 18 Ejemplo petición JSON para programar una extracción diaria nodo

3.6.8. Servicio de Nodo semanal

Comprueba la existencia del nodo el cual se quiere programar, luego se crea la programación del extractor un día de la semana a una hora especificada por el usuario.

```
{
  "extractorID": "extractorID",
  "nodeName": "Nodo",
  "dayOfWeek": "Miercoles",
  "atHour": 5,
  "atMinute": 0
}
```

Ilustración 19 Ejemplo petición JSON para programar una extracción semanal nodo

3.6.9. Servicio de Nodo mensual

Comprueba la existencia del nodo el cual se quiere programar, luego se crea la programación del extractor un día X del mes a una hora especificada por el usuario.

```
{
  "extractorID": "extractorID",
  "nodeName": "Nodo",
  "dayOfMonth": 15,
  "atHour": 5,
  "atMinute": 0
}
```

Ilustración 20 Ejemplo petición JSON para programar una extracción mensual nodo

4. SISTEMA: INTEGRACIÓN DE LOS SERVICIOS INCORPORADOS EN LA API TECO EN UNA APLICACIÓN WEB.

4.1. AMBIENTE DE INGENIERÍA DE SOFTWARE

4.1.1. Metodología de Desarrollo

El proyecto a desarrollar será abordado bajo una metodología evolutiva e incremental, esto porque, se pretende ir desarrollando partes funcionales de la aplicación (Incremental), lo que serían nuestros incrementos, los cuales se irán sumando para completar el producto final. Los investigadores del grupo SOMOS, nuestros clientes y usuarios, pueden ir probando y utilizando lo desarrollado para luego obtener una retroalimentación, lo cual podría conllevar pequeñas correcciones o algún nuevo requerimiento (Evolutiva).

Lo fundamental de la metodología es la creación de versiones de la aplicación las cuales se tomarán como la base al siguiente incremento en donde se pueden añadir nuevas características e ir mejorando las que ya existen.

4.1.2. Estándares de documentación

- **Estándar para los requerimientos funcionales:** Adaptación basada en IEEE Software requirements Specifications Std 830-1998.

4.1.3. Técnicas y notaciones

- **Lenguaje de unificado y modelado (UML):** Se utiliza para una mejor comprensión mediante notación gráfica, en el cual se describirán los casos de uso para así visualizar las operaciones que podrá realizar el usuario en el sistema y disminuir la ambigüedad.
- **Modelo Entidad-Relación:** Es un modelo lógico que muestra cómo interactúan las entidades entre sí.

4.1.4. Herramientas de apoyo al desarrollo de software que serán utilizadas

- **Sublime Text 3:** Editor de texto y de código fuente.
- **PHP:** Lenguaje de programación de código del lado del servidor.
- **Insomnia:** Herramienta para la crear solicitudes HTTP, permite realizar debug a APIs.
- **GIT:** Software para el control de versiones.
- **SourceTree:** Cliente de GIT, facilita la interacción con repositorios GIT mediante interfaz gráfica.
- **Lucidchart:** Herramienta de diagramación basada en la web.
- **WampServer:** Un entorno de desarrollo web para Windows con el que se pueden llevar acabo aplicaciones web, en este caso con Apache y PHP.
- **Navegador Web:** Aplicación que permite el acceso a la Web, interpretando información de distintos tipos de archivos y sitios web para que puedan ser visualizados.

4.1.5. Hardware para el desarrollo de software

Para el desarrollo de este proyecto se utiliza un equipo con las siguientes características:

<i>Sistema Operativo</i>	Windows 10 x64
<i>Procesador</i>	Intel Core i5-6200U
<i>Memoria RAM</i>	8GB de RAM

Tabla 1 Hardware utilizado para el desarrollo

4.2. ESPECIFICACIÓN DE REQUERIMIENTOS

4.2.1. Objetivo General del Software

Implementación de interfaces gráficas para la API TECO que faciliten y agilicen las extracciones de los corpus necesarios para el análisis de subjetividad de texto.

4.2.2. Objetivos Específicos del Software

- Facilitar la construcción de peticiones desde un cliente a los servicios de la API.
- Felicitar la recuperación de los datos que son extraídos.
- Reducir tiempo en la creación y ejecución de extractores.
- Mejorar experiencia de usuario al momento de utilizar los servicios de la API TECO.

4.2.3. Descripción global del producto

4.2.3.1. Interfaz de Hardware

El sistema no requiere la configuración de dispositivos específicos, sólo se necesita la configuración estándar para los periféricos básicos, tales como:

- Entrada estándar: teclado, mouse.
- Salida estándar: monitor.
- Tarjeta de red: wifi o ethernet.

4.2.3.2. Interfaz de Software

El sistema requiere relacionarse con la interfaz de programación de aplicaciones llamada “Text Extraction Community - API Service” las cual cuenta con servicios para la extracción de texto.

4.2.3.3. Interfaces de Comunicación

Para la interacción del cliente con el software es necesario contar con los siguientes protocolos de comunicación:

- **Protocolo TCP/IP:** Este protocolo representa todas las reglas de comunicación para internet y basa su funcionamiento en las direcciones IP. Esto permite una comunicación fiable y segura extremo a extremo.

- **HTTP:** El significado de esta sigla es *HyperText Transfer Protocol* (Protocolo de transferencia de hipertexto) es el método más común para intercambiar información en la web. La comunicación entre el servidor y el cliente se realizará a través del puerto TCP: 80 para el tráfico HTTP.

4.2.4. Requerimientos funcionales del sistema

ID	Nombre	Descripción
RF01	Login	El sistema permite ingresar al sistema mediante un usuario y contraseña.
RF02	Crear Cuenta	El sistema permite registrar nuevos usuarios.
RF03	Activar Cuenta	El sistema permite activar las cuentas de usuario para poder ser utilizadas.
RF04	Registrar nuevo extractor	El sistema permite a los usuarios crear sus propios extractores.
RF05	Ejecutar extractores públicos y propios	El sistema permitirá utilizar los extractores propios de un usuario y así como otros extractores públicos.
RF06	Programar ejecución de extractores	El sistema permitirá programar extractores para su ejecución automática para una posterior recuperación de datos
RF07	Ejecutar extracción desde Twitter	El sistema permitirá ingresar consultas que serán extraída desde Twitter
RF08	Generar CSV	El sistema permite generar un archivo CSV con los contenidos obtenido por la ejecución de extractores

Tabla 2 Requerimientos funcionales del sistema

4.2.5. Requerimientos no funcionales

ID	Nombre	Descripción
<i>RNF01</i>	Soporte de navegadores (Portabilidad)	El sistema debe funcionar en múltiples navegadores tales como: Mozilla Firefox, Google Chrome, etc.
<i>RNF02</i>	Patrón de diseño (Mantenibilidad)	El sistema está diseñado bajo el patrón estructural Adapter.
<i>RNF03</i>	Arquitectura tres capas (Facilidad de análisis)	El sistema de estar construido con una arquitectura de tres capas: Cliente, Servidor y datos.
<i>RNF04</i>	Diseño web responsivo	El sistema debe ser capaz de adaptar su apariencia según el dispositivo en donde se utilice.

Tabla 3 Requerimientos no funcionales

4.2.6. Interfaces externas de entrada

ID	Nombre	Detalles
<i>IE01</i>	Datos Extractor	Nombre URL Descripción Compartir Nodos
<i>IE02</i>	Datos Nodo	Nombre Contenedor Descripción Elementos
<i>IE03</i>	Datos Elemento	Etiqueta Selector
<i>IE04</i>	Consulta Twitter	Consulta

Tabla 4 Interfaces externas de entrada

4.2.7. Interfaces externas de salida

ID	Nombre	Detalles	Medio de salida
IS01	Extractores	Nombre Descripción URL	Pantalla
IS02	Información de usuario	ID Nombre Usuario Email Fecha Registro	Pantalla
IS03	Información Obtenida de Extractores	Nombre Extractor Nombre Nodo Datos	Pantalla Archivo CSV
IS04	Información Extraída desde Twitter	Fecha Nombre Cuenta Usuarios Twitter Texto Respuestas Retweet Favorito	Pantalla Archivo CSV

Tabla 5 Interfaces externas de salida

4.3. ESTUDIO DE FACTIBILIDAD.

El estudio de factibilidad nos permitirá comprobar la disponibilidad de diferentes recursos, tanto como la infraestructura y capacidades técnicas, que son necesarios para lograr cumplir con los objetivos que se han propuestos en este proyecto. Los aspectos para considerar en el análisis de factibilidad serán los 3 estudios que se describen a continuación:

- Factibilidad técnica
- Factibilidad económica
- Factibilidad operacional

4.3.1. Factibilidad técnica

Para llevar a cabo el desarrollo del proyecto se han considerado la evaluación de tres tipos de recursos, los cuales son recursos humanos, hardware de desarrollo y herramientas de desarrollo.

4.3.1.1. Recursos humanos

El desarrollo del proyecto está a cargo del alumno Jorge Ariel Vallejos Prado memorista de la carrera Ingeniería de Ejecución en Computación e Informática.

Capacidades Técnicas del Desarrollador

- Conocimiento en PHP
- Conocimiento en JavaScript
- Conocimiento en HTML y CSS

4.3.1.2. Recursos de Hardware

- Computador Portátil
- Conexión a Internet

4.3.1.3. Herramientas de Desarrollo

Los recursos a utilizar se encuentran disponibles para libre descarga,

Tipo	Nombre	Versión
<i>Lenguaje de programación</i>	PHP	5.6.35
<i>Lenguaje de programación biblioteca</i>	JavaScript	-----
	Jquery	3.1.1
<i>Entorno de desarrollo</i>	WampServer	3.1.3
<i>Servidor web</i>	Apache	2.4.33
<i>framework</i>	Laravel	5.4

Tabla 6 Herramientas de Desarrollo

4.3.2. Factibilidad Económica

No se requiere inversión por parte de la Dirección de Informática o del alumno en cuanto:

- El alumno memorista dispone de equipo y conexión a Internet para el desarrollo del proyecto.
- La universidad cuenta con servidores web.
- La universidad del Bío-Bío cuenta con una conexión a Internet
- Dado que este proyecto se desarrolla como trabajo de título, el coste de software es cero, ya que se desarrolla con un framework de código abierto y el costo de implementación es asumido por el memorista, desarrollador del proyecto.

4.3.3. Factibilidad Operativa

Puesto que el grupo SOMOS cuentan con los conocimientos necesarios para la creación de extractores y análisis de texto, se asume que el uso del software por parte de los usuarios no conllevara complicaciones, además de facilitar la recuperación de información que serán los corpus para sus análisis investigativos.

4.3.4. Conclusión Factibilidad

Se asume que el desarrollo del sistema es factible, ya que se asumen que los usuarios finales cuentan con las habilidades necesarias para hacer correcto uso al sistema, por otra parte los costos asociados a su desarrollo son relativamente bajos, debido a que el desarrollo del sistema está a cargo de un alumno memorista no existen gastos de por medio en cuanto al pago por el desarrollo ni adquisición de hardware para el sistema, además las herramientas de desarrollo a utilizar son herramientas de uso libre por lo cual tampoco se incurren en gastos en ello.

4.4. ANÁLISIS

4.4.1. Diagramas de casos de Uso

4.4.1.1. Actores

Usuario: El usuario es el actor que crea y ejecuta extractores para descargar texto de páginas webs de los cuales se obtienen los corpus sobre los cuales se realizan los análisis diferentes análisis sobre ellos

4.4.1.2. Caso de Uso y Descripción

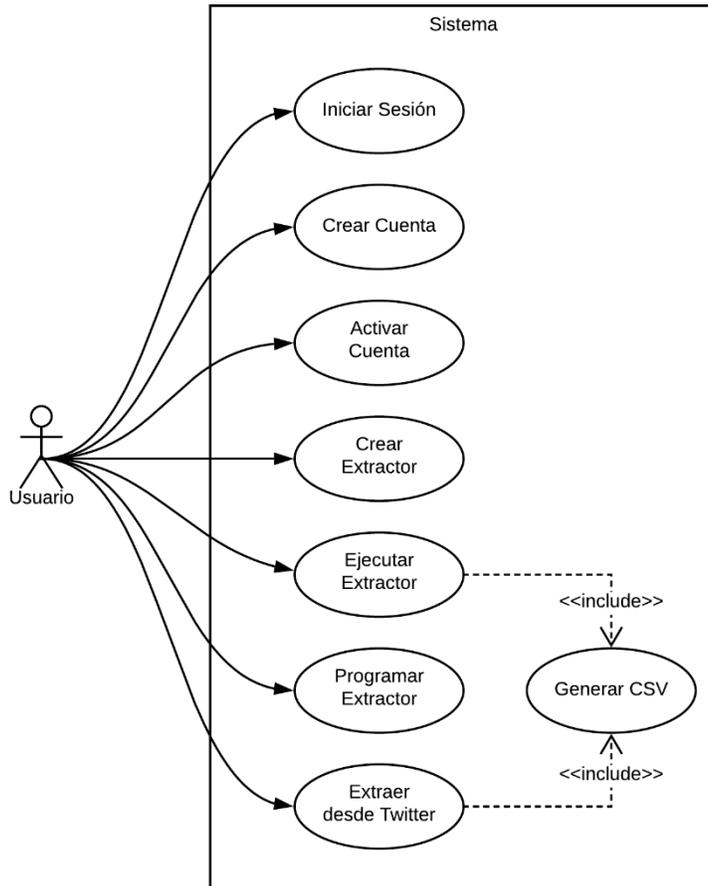


Ilustración 21 Caso de Uso

4.4.1.3. Especificación de casos de uso

Caso de Uso	Iniciar sesión
Actor/es implicado/s	Usuario
Descripción	Permite al usuario ingresar al sistema.
Precondición	<ul style="list-style-type: none"> • No haber ingresado al sistema o no tener sesión iniciada. • Estar registrado en la API TECO.
Postcondición	<ul style="list-style-type: none"> • Ingreso al sistema con sus opciones habilitadas.
Curso normal de eventos	
Acción del Actor	Respuesta del sistema
1.- El usuario ingresa sus datos de usuario y contraseña en la página principal.	2.- El sistema valida, si son correctos, los datos del usuario ingresados a la API TECO.
	3.- El sistema verifica los datos enviados e ingresa al sistema
Curso alternativo de eventos	
1.b.- El usuario ingresa datos erróneos o inexistentes.	2.b.- El sistema despliega un error de datos inválidos.

Tabla 7 Caso de uso: Iniciar Sesión

Caso de Uso	Crear Cuenta
Actor/es implicado/s	Usuario
Descripción	Permite registrar un nuevo usuario en el sistema.
Precondición	<ul style="list-style-type: none"> No estar registrado en el sistema.
Postcondición	<ul style="list-style-type: none"> Usuario registrado en sistema, pero aún inactivo.
Curso normal de eventos	
Acción del Actor	Respuesta del sistema
1.- El usuario ingresa sus datos de usuario y contraseña en la página principal	2.- El sistema despliega un mensaje con instrucciones de activación.
Curso alterno de eventos	
1.b.- El usuario ingresa datos erróneos o ya existentes.	2.b.- El sistema despliega un error de datos inválidos.

Tabla 8 Caso de Uso: Crear Cuenta

Caso de Uso	Activar Cuenta
Actor/es implicado/s	Usuario
Descripción	Permite activar la cuenta de usuario para utilizar los servicios de extracción.
Precondición	<ul style="list-style-type: none"> • Estar registrado en el sistema. • Tener código de activación.
Postcondición	<ul style="list-style-type: none"> • Cuenta activada.
Curso normal de eventos	
Acción del Actor	Respuesta del sistema
1.- El usuario ingresa sus datos de usuario y código de activación en la página principal.	2.- El sistema despliega mensaje de activación exitosa.
Curso alterno de eventos	
1.b.- El usuario ingresa datos o código erróneo.	2.b.- El sistema despliega un error de datos inválidos.

Tabla 9 Caso de uso: Activar Cuenta

Caso de Uso	Crear Extractor
Actor/es implicado/s	Usuario
Descripción	Permite ingresar un nuevo extractor al sistema
Precondición	<ul style="list-style-type: none"> • Iniciar sesión en el sistema. • Extractor disponible
Postcondición	<ul style="list-style-type: none"> • CSV disponible para descarga
Curso normal de eventos	
Acción del Actor	Respuesta del sistema
	1.- El sistema despliega formulario para crear extractor.
2.- El usuario los datos de la web a extraer.	3.- El sistema permite ingresar un nuevo nodo.
4.- EL usuario ingresa los datos del nodo	5.- El sistema permite la verificación del nodo.
6.- EL usuario confirma y guarda extractor.	

Tabla 10 Caso de Uso: Crear Extractor

Caso de Uso	Ejecutar Extractor
Actor/es implicado/s	Usuario
Descripción	Permite ingresar un nuevo extractor al sistema
Precondición	<ul style="list-style-type: none"> • Iniciar sesión en el sistema.
Postcondición	<ul style="list-style-type: none"> • Extractor disponible para su ejecución
Curso normal de eventos	
Acción del Actor	Respuesta del sistema
	1.- El sistema activa la descarga de archivo CSV.
2.- El usuario presiona el botón generar CSV.	3.- El sistema genera archivo en formato CSV y ejecuta la descarga.

Tabla 11 Caso de Uso: Ejecutar Extractor

Caso de Uso	Programar Extractor
Actor/es implicado/s	Usuario
Descripción	Permite realizar una extracción programada de un extractor
Precondición	<ul style="list-style-type: none"> • Iniciar sesión en el sistema.
Postcondición	<ul style="list-style-type: none"> • Se pueden rescatar los datos extraídos por la programación.
Curso normal de eventos	
Acción del Actor	Respuesta del sistema
1.- Ingresar a vista mis extractores	2.- El sistema despliega los extractores disponibles
2.- El usuario selecciona que tipo de programación desea crear.	3.- El sistema despliega el formulario de programación.
4.- EL usuario completa la información de acuerdo al tipo de programación seleccionada	5.- El sistema recoge y guarda los datos de la programación
Curso alternativo de eventos	
	5.b.- El sistema despliega mensaje de error si ya se programó una extracción del mismo tipo para el extractor seleccionado

Tabla 12 Caso de Uso: Programar Extractor

Caso de Uso	Extraer desde Twitter
Actor/es implicado/s	Usuario
Descripción	Permite ingresar una consulta la cual permitirá obtener contenido desde Twitter que coincidan con lo ingresado
Precondición	<ul style="list-style-type: none"> • Iniciar sesión en el sistema.
Postcondición	<ul style="list-style-type: none"> • CSV disponible para descarga
Curso normal de eventos	
Acción del Actor	Respuesta del sistema
	1.- El sistema despliega formulario para ingresar una consulta
2.- El usuario ingresa la consulta que desea buscar.	3.- El sistema recoge todas las coincidencias y las despliega en pantalla
	4.- El sistema activa la descarga de archivo CSV.

Tabla 13 Caso de Uso: Extraer desde Twitter

Caso de Uso	Generar CSV de textos
Actor/es implicado/s	Usuario
Descripción	Permite generar un archivo con formato CSV de los textos extraídos de la ejecución de algún extractor
Precondición	<ul style="list-style-type: none"> • Iniciar sesión en el sistema.
Postcondición	
Curso normal de eventos	
Acción del Actor	Respuesta del sistema
	1.- El sistema despliega la opción para generar el CSV.
2.- El usuario selecciona la opción generar CSV.	3.- El sistema genera archivo en formato CSV y ejecuta la descarga.

Tabla 14 Caso de Uso: Generar CSV de Textos

4.4.1.4. Trazabilidad de casos de uso

En la siguiente tabla se muestra la relación existente entre los requerimientos funcionales con los casos de usos, permite visualizar que requerimiento queda cubierto por un caso de uso.

	RF01	RF02	RF03	RF04	RF05	RF06	RF07	RF08
CU1	X							
CU2		X						
CU3			X					
CU4				X				
CU5					X			
CU6						X		
CU7							X	
CU8								X

Tabla 15 Trazabilidad de casos de uso

4.4.2. Modelamiento de Datos

El modelamiento de datos permite identificar los datos que existen en la base de datos y determinar la forma en que se relacionan unos con otros, todo esto bajo una base de datos relacional. Dado a que se está trabajando con una API externa la cual no cuenta con una base de datos relacional por lo que no existe un diagrama que pueda describirla, puesto que no cuenta con una estructura fija permitiendo obtener mayor escalabilidad de datos, sin embargo, la interacción de la API hacia el usuario es realizada a través de la notación JSON se puede realizar una representación del esquema más importante que será guardado en la base de datos, y también una adaptación de este diagrama a uno de Entidad-Relación.

```

Usuario {
  id : string, 24 caracteres (autogenerado)
  correoElectronico : string, 8-50 caracteres
  usuario : string, 4-50 caracteres
  contraseña : string, 8-30 caracteres (hash encriptado 60 caracteres)
  extractores {
    "Nombre del Extractor" { string, 4-50 caracteres
    url : string, 16-255 caracteres
    descripcion : string, 4-255 caracteres
    nodosExtractores {
      "Nombre del Nodo" { string, 4-50 caracteres
      Descripción : string, 4-255 caracteres
      Contenedor : string, 4-255 caracteres
      ElementosExtractores {
        Etiqueta . string, 4-30 caracteres
        Ruta : string, 1-255 caracteres
      }
    }
  }
}

```

Ilustración 22 Representación JSON de la API

El siguiente diagrama muestra una adaptación del JSON presentado anteriormente a un diagrama Entidad-Relación.

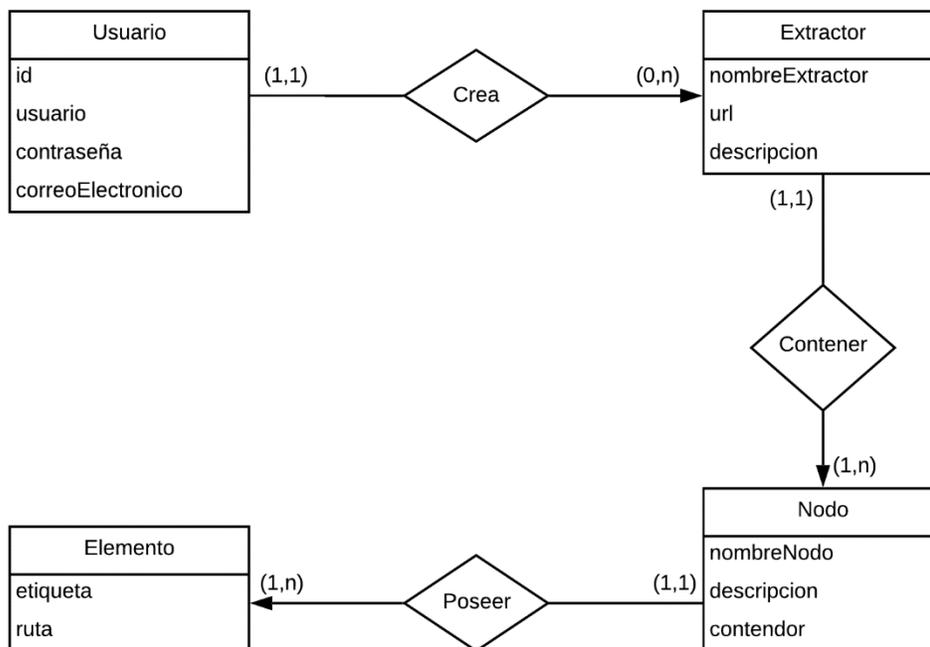


Ilustración 23 Adaptación JSON a Entidad-Relación

4.5. DISEÑO Y CONSTRUCCIÓN

4.5.1. Diseño de arquitectura funcional

La aplicación desarrollada en este proyecto estará basada en PHP, en la cual se utilizará patrón de diseño de software Adapter, que permite crear una envoltura o adaptador para la comunicación con la API. Todas las operaciones de la aplicación que necesiten trabajar con la API se deben centralizar en el adaptador, el cual realiza la "envoltura" de los servicios concentrando en el las clases más importantes o transversales.

4.5.2. Diseño interfaz y navegación

A continuación, se presenta una serie de mockups de la interfaz gráfica de usuario que serán incorporadas en el proyecto.

La siguiente imagen muestra la vista de ingreso al sistema.

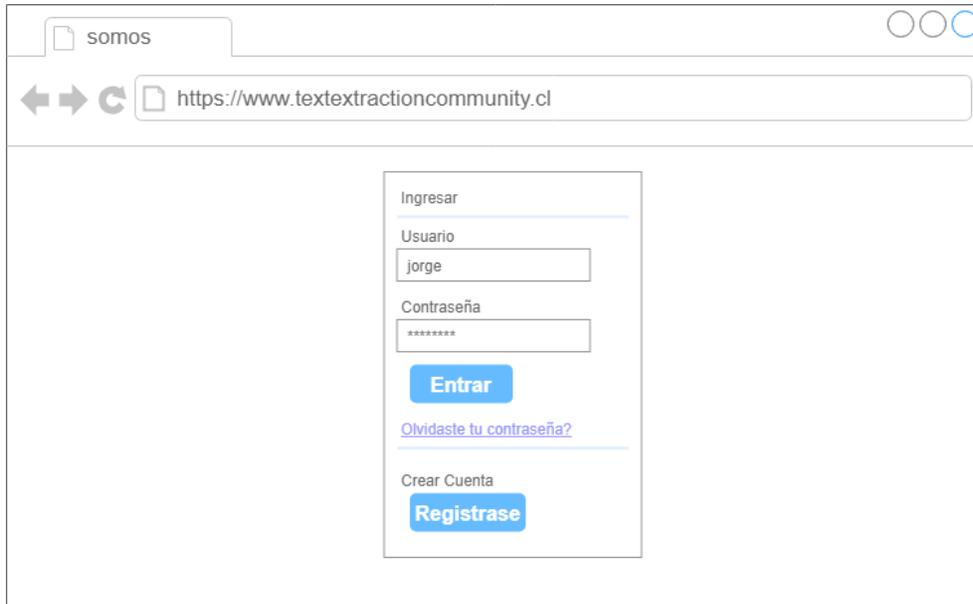


Ilustración 24 Vista ingreso al sistema

La siguiente imagen muestra la pantalla en la cual se podrán visualizar los extractores propios del usuario que ha ingresado al sistema.

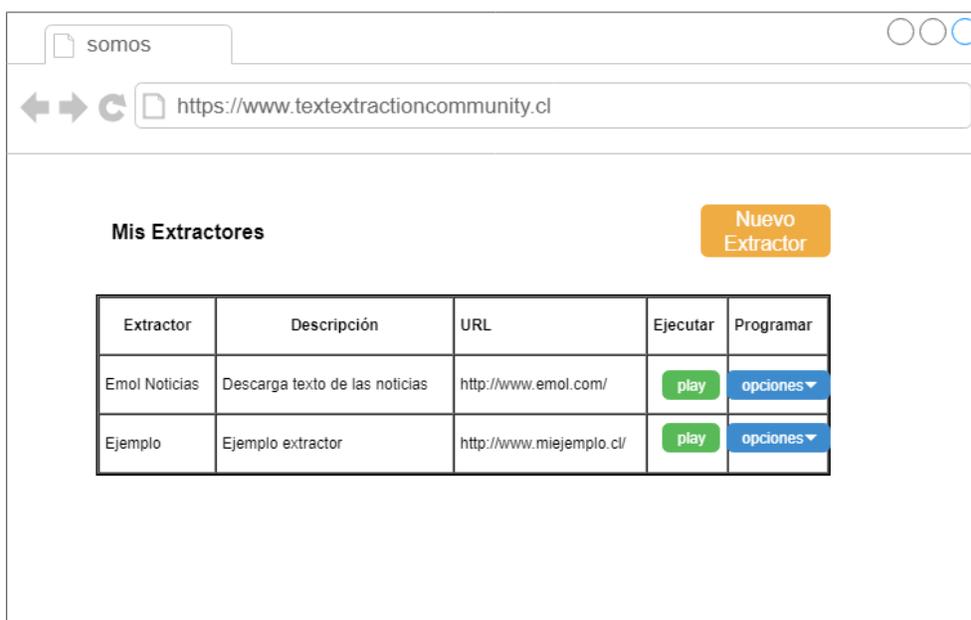


Ilustración 25 Vista mis extractores

La siguiente imagen muestra la pantalla en la cual se podrán visualizar los extractores que fueron compartidos en modo público.

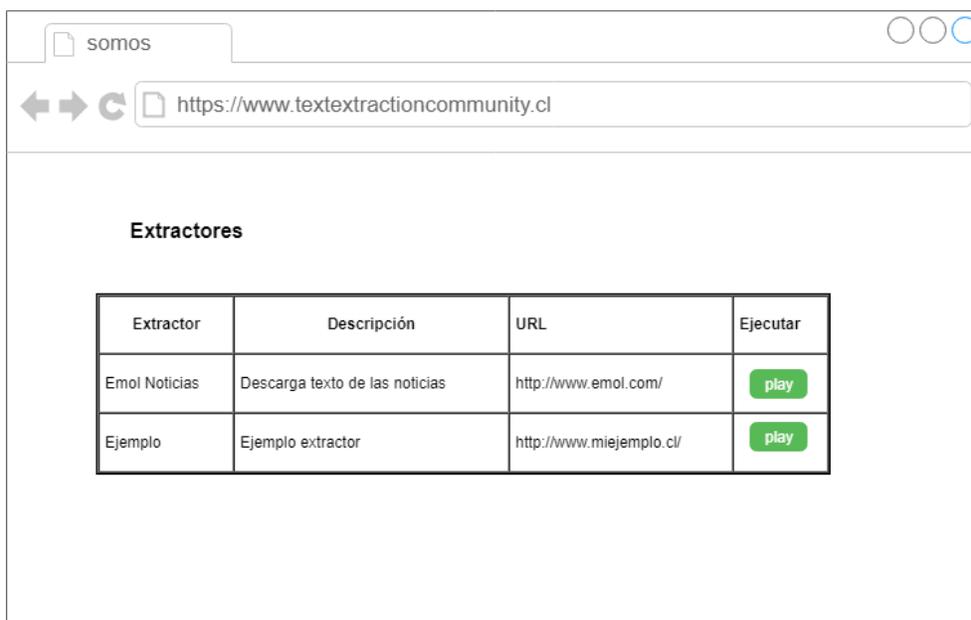


Ilustración 26 Vista extractores públicos

La siguiente imagen muestra la pantalla en la cual se visualizarán los resultados obtenidos al ejecutar un extractor.

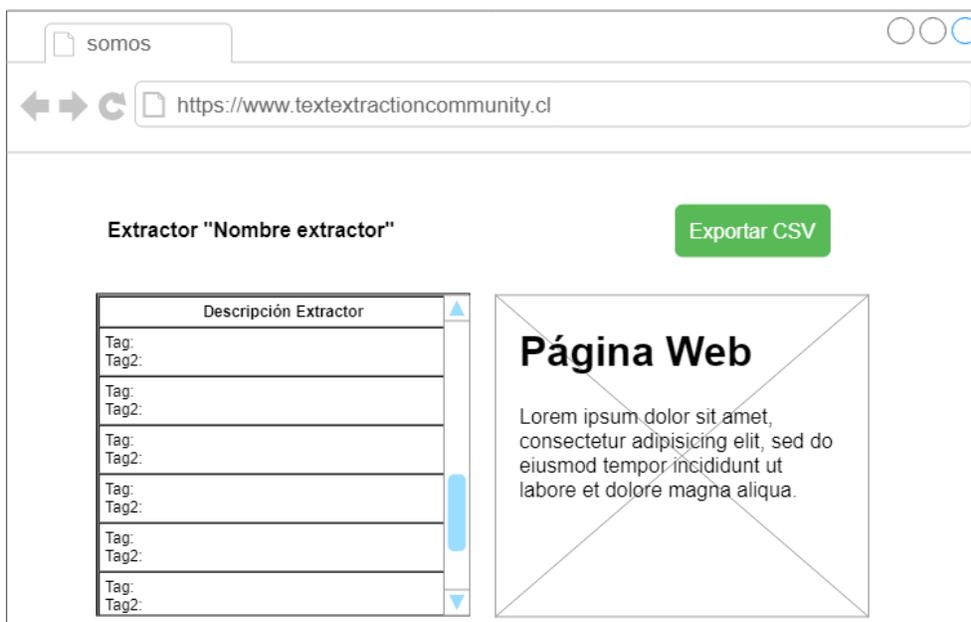


Ilustración 27 Vista resultados de ejecución

La siguiente imagen muestra la vista en la cual se pueden ingresar consultas para buscar Tweets según algún criterio de búsqueda.

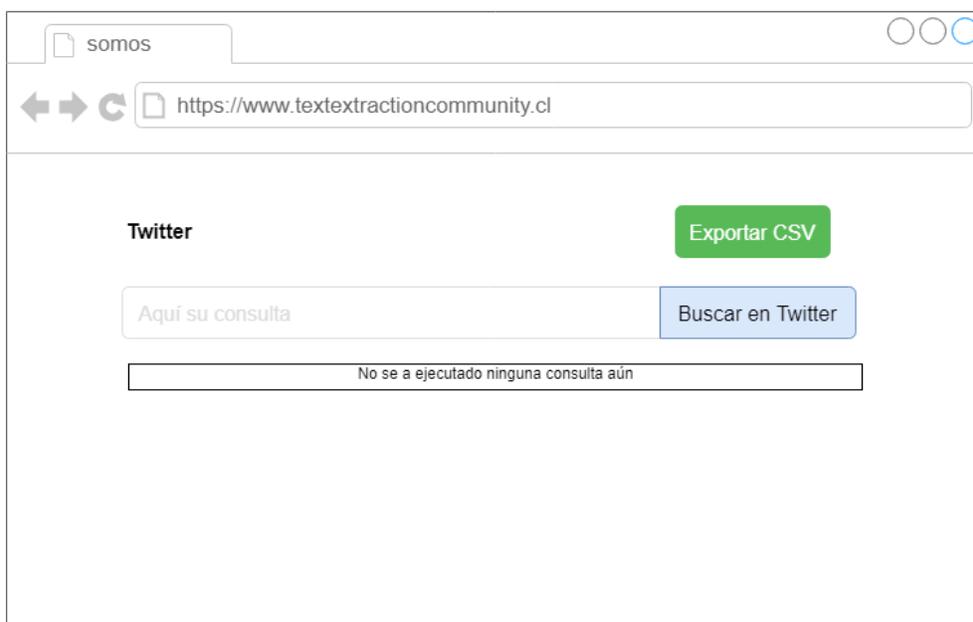


Ilustración 28 Vista búsqueda en Twitter

La siguiente imagen muestra la pantalla en la cual se podrá realizar la creación de un nuevo extractor.

The image shows a web browser window with a single tab titled 'somos'. The address bar contains the URL 'https://www.textextractioncommunity.cl'. The main content area displays a form titled 'Nuevo Extractor'. The form consists of several input fields and buttons:

- Página Web:** A text input field containing 'http://www.ejemplo.cl/'.
- Nombre:** A text input field containing 'Ejemplo'.
- Descripción:** A text input field containing 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna ...'.
- Nodo (Node):** A sub-section containing:
 - Nombre:** A text input field containing 'Noticias'.
 - Contenedor:** A text input field containing '.classNoticia.ultimoejemplo'.
 - Descripción:** A text input field containing 'Esto es un nodo'.
- Elemento (Element):** A sub-section containing:
 - Etiqueta:** A text input field containing 'titulo'.
 - Selector:** A text input field containing 'h1'.

At the bottom of the form, there are three buttons: a green button labeled 'Nuevo elemento', a blue button labeled 'Nuevo Nodo', and an orange button labeled 'Guardar'.

Ilustración 29 Vista nuevo extractor

La siguiente imagen muestra la pantalla para la recuperación de datos de las extracciones que han sido programadas.

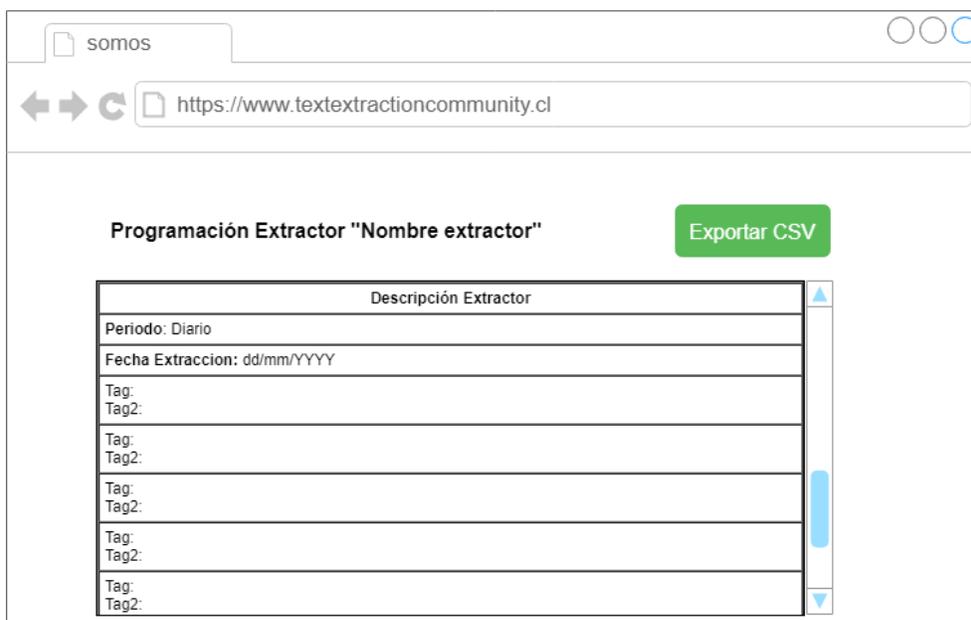


Ilustración 30 Vista resultado extracción programada

5. PRUEBAS

5.1. ELEMENTOS DE PRUEBA

Elementos de prueba	Descripción
Extraer desde Twitter	El sistema permitirá extraer contenido desde Twitter mediante una consulta ingresada.
Generar extractor	El sistema permitirá la creación de un nuevo extractor.

5.2. ESPECIFICACIÓN DE PRUEBAS

Características de la prueba	Funcionalidad.
Nivel de prueba	Aceptación.
Objetivo de la prueba	Verificar que se cumplan los requisitos del sistema; que este sea capaz de rescatar información de fuentes externas y ayudar al usuario en la creación de extractores.
Enfoque para la definición de los casos de prueba	Caja negra.
Técnicas para la definición de los casos de prueba	Pruebas manuales.
Criterio de cumplimientos	Que el sistema sea capaz de entregar respuestas satisfactorias en base a parámetros ingresados por el usuario.

5.3. RESPONSABLE DE LAS PRUEBAS

El responsable de las pruebas es el autor del informe, el memorista Jorge Vallejos Prado.

5.4. DETALLE DE LAS PRUEBAS

- **Elemento de prueba:** Extraer desde Twitter.

ID	1	2	3
Criterio de Búsqueda	<i>#medidoresinteligentes</i>	<i>"Sebastián Piñera"</i> <i>filter:links</i>	<i>terremoto until:2010-03-05</i>
Resultados Esperado	Tweets que contengan el hashtag <i>#medidoresinteligentes</i>	Tweets que contengan exactamente <i>Sebastián Piñera</i> e inserten una URL.	Tweets que contengan "terremoto" y sean enviados hasta la fecha "2010-12-27" (año-mes-día)
Cantidad de Resultados	518	500	520
Usuario que publica	cumple	cumple	cumple
Fecha y Hora de publicación	cumple	cumple	cumple
Cantidad de respuestas	cumple	cumple	cumple
Cantidad de favoritos	cumple	cumple	cumple
Cantidad Retweet	cumple	cumple	cumple
Textos contra Criterio de búsqueda	cumple	cumple	cumple

Tabla 16 Prueba Extraer desde Twitter

- Primer caso se decide realizar búsqueda de contenido mediante el operador # bajo el hashtag “meidoresinteligentes”. Para ejecutar la búsqueda se accede al menú lateral izquierdo “Twitter”, en parte superior se selecciona el operador a ejecutar, en este caso operador # para realizar la búsqueda de tweets que contengan *#medidoresinteligentes*, obteniendo como resultado una cantidad de 520 tweets asociados a nuestra búsqueda. Para el análisis se recogen 100 de los resultados, de los cuales todos cumplen con el criterio de búsqueda seleccionado.
- Segundo caso se decide realizar búsqueda de contenido mediante el operadores “ ” y *filter* bajo el criterio *"Sebastián Piñera" filter:links*. Para ejecutar la búsqueda se accede al menú lateral izquierdo “Twitter”, en parte superior se selecciona el operador

a ejecutar, en este caso dos operadores combinados, el operador “ ” se utiliza para realizar la búsqueda de tweets que contengan exactamente las palabras que se encuentran entre las comillas, por otra parte el operador *filter* es utilizado para filtrar bajo algún otro parámetro, en este caso *links*, el cual nos permite obtener Tweets que contengan URLs, por lo cual en esta búsqueda se recuperan Tweets que contengan exactamente “Sebastián Piñera” y que contengan algún URL. Tras la ejecución se obtuvieron como resultado una cantidad de 500 tweets asociados a nuestra búsqueda. Para el análisis se recogen 100 de los resultados, de los cuales todos cumplen con el criterio de búsqueda seleccionado, conteniendo “Sebastián Piñera” y conteniendo una URL.

- Tercer caso se decide realizar búsqueda de contenido mediante el operador *until* bajo el criterio *terremoto until:2010-03-05*. Para ejecutar la búsqueda se accede al menú lateral izquierdo “Twitter”, en parte superior se selecciona el operador a ejecutar, el operador *until* se utiliza para realizar la búsqueda de tweets que hayan sido publicados hasta la fecha indicada. En este caso se realizó la búsqueda de Tweets que contengan la palabra *terremoto* y hayan sido publicado hasta la fecha *2010-03-05(año-mes-día)*. Tras la ejecución se obtuvieron como resultado una cantidad de 520 tweets asociados a nuestra búsqueda. Para el análisis se recogen 100 de los resultados, de los cuales todos cumplen con el criterio de búsqueda seleccionado, *terremoto until:2010-03-05*.

• Elemento de prueba: Generar extractor.

ID	1	2
Página Ingresada	https://www.biobiochile.cl/	https://www.pagina7.cl/
Contenedores obtenidos	59 posibles contenedores	29 posibles contenedores
Contenedor utilizado	<code>.section.section-blogs-bbcl</code> <code>.article.pb-2.mb-2</code>	<code>d-block .actualidad-noti</code>
Selectores obtenidos	7 posibles Selectores	7 posibles Selectores
Selector utilizado	<code>.article-title</code>	<code>.titulo</code>
Respuesta de extracción de prueba	<ul style="list-style-type: none"> • Crítica: las 5 polémicas escenas de Game of Thrones 8 que han humillado a las mujeres • Ya vimos la nueva apuesta juvenil de Netflix "The Society" y este es nuestro veredicto • titulo: Los extremos abusos del conservadurismo religioso 	<ul style="list-style-type: none"> • 18 mayo, 2019 20:45 hrs. Carabineros confirmó diligencias por masiva clonación de tarjetas bancarias en Santiago • 18 mayo, 2019 20:05 hrs. Comiendo sobras de papas fritas: encontraron a niño autista abandonado en local de comida rápida • 18 mayo, 2019 19:17 hrs. Familia de niño muerto en viaje deportivo demanda a Scotiabank: pide indemnización de \$900 millones • 18 mayo, 2019 15:31 hrs. Intentó quemarla vida: hombre fue detenido tras prender fuego a su casa con su pareja adentro

Tabla 17 Prueba Generar Extractor (a)

ID	3	4
Página Ingresada	https://www.latercera.com/	https://www.elmostrador.cl/
Contenedores obtenidos	1 posible contenedor	12 posibles contenedores
Contenedor utilizado	.container .row	.col-xs-12.col-sm-12.col-md-12.item .contenido-entrada
Selectores obtenidos	68 posibles Selectores	5 posibles selectores
Selector utilizado	.fav-icon.fi-lt .ltpicto-fav-tendencia	.col-xs-9.col-sm-9.col-md-9
Respuesta de extracción de prueba	Sin Resultado	<ul style="list-style-type: none"> • Elecciones en India: Las más grandes y complejas del mundo democrático por Mladen Yopo • ¿Cuándo se acaba el gobierno? por Axel Callis • Los loteos de Moreno huelen mal por Maya Fernández • ¿Hasta cuándo ganan las Isapres? por María Constanza Galleguillos • Superclásico: todo puede pasar por Pablo Flamm • Siete claves transformadoras de la Universidad de Concepción en su centenario por Esteban Valenzuela • Salud y migrantes: preocupante retroceso por Izkia Siches y Matías Libuy • Elegir entre economía estancada y mayor globalización por Noah Smith

Tabla 18 Prueba Generar Extractor (b)

- En cada uno de los casos anteriores se utiliza el creador de extractores asistido, para realizar las pruebas se utilizaron páginas de noticias nacionales, específicamente en este caso las páginas “biobiochile.cl”, “pagina7.cl”, “latercera.com” y “elmostrador.cl”, las pruebas consistían en crear extractores de contenido de cada uno de estos sitios en los cuales el sistema detectara automáticamente posibles elementos contenedores y elementos selectores, los cuales tras una ejecución del extractor permitieran la obtención de contenido de los sitios mencionados anteriormente, sin embargo en el caso del sitio “latercera.com”, el resultado no fue satisfactorio, pues no fue posible obtener contenido tras la combinación del contenedor con el selector, cabe mencionar que la detección de contenedores y selectores esta basado en las clases que posee una web, es por esto y por la forma en que un sitio esta construido varia de uno a otro, que la detención de elementos extraíbles, no es asegurada en un 100%.

5.5. CONCLUSIONES DE LAS PRUEBAS

El sistema responde de forma satisfactoria ante las extracciones de contenido desde Twitter y permite un correcto uso de los diferentes operadores que le dan la opción al usuario de cambian los criterios de búsquedas. Por otra parte, la creación de extractores se comporta de la forma esperada, entregando al usuario elementos que podrían ser extraídos desde una página web, recordar que esta función no es totalmente precisa puesto que diversos factores externos, principalmente la forma en que se construye un sitio, influirá en la detección de los elementos que pueden ser extraídos.

6. CONCLUSIONES

En relación a lo antes expuesto, se puede determinar que cada objetivo que ha sido planteado dentro del desarrollo del proyecto ha logrado un grado de aceptación y funcionalidad que permite su utilización dentro de lo planteado y de su funcionalidad principal que es la extracción de contenido desde páginas webs puede ser utilizado por los usuarios.

Por otra parte las herramientas de desarrollo y lenguaje de programación utilizado han logrado aportar y adaptarse de forma correcta al desarrollo de la aplicación, aunque de un principio se debió invertir parte de tiempo en cómo realizar el consumo de una API externa dentro del framework Laravel que fue utilizado en el desarrollo de este proyecto, luego de esto la adaptación de la capa intermedia que conecta la API con la aplicación desarrollada, fue la que juega el papel principal en el desarrollo, lo cual con las herramientas utilizadas se puede obtener un buen comportamiento de esta.

Sin duda el desarrollo de este proyecto fue una gran forma de aplicar y consolidar los conocimientos adquiridos a lo largo de la duración de la carrera, sin embargo, también fue una gran instancia para adquirir una nueva experiencia y nuevos conocimientos para fortalecer las capacidades como profesional con las cuales se egresa de la carrera, además de ser una instancia en la cual se puede demostrar la capacidad que se tiene como profesional para afrontar un problema.

A nivel personal el proyecto realmente implicó un desafío a superar, puesto que se debió trabajar de una forma, que en mis años de estudio no se contempló plenamente, el hecho de trabajar una API externa implica trabajar con una parte que realmente ignoras internamente su funcionamiento y solo conoces su interfaz, que es la que te permite la comunicación con esta, es por esto que el entender cómo funciona la API y cómo se puede realizar un intercambio de datos con esta, conlleva un tiempo de análisis y conocimiento para poder integrar de forma correcta dentro de otro proyecto. El desconocer esto y tener una baja experiencia significó ver el proyecto con un grado de complejidad mayor pero siempre superable, y de seguro el aprendizaje adquirido será de gran valor para mi desarrollo como profesional en el área de la informática.

7. BIBLIOGRAFÍA

- [1] Rouse, M. (2014). What is application program interface (API)? - Definition. Retrieved October 1, 2018, from <https://searchmicroservices.techtarget.com/definition/application-program-interface-API>
- [2] HTTP Requests | Codecademy. (n.d.-e). Retrieved October 8, 2018, from <https://www.codecademy.com/articles/http-requests>
- [3] Guzzle, PHP HTTP client — Guzzle Documentation. (n.d.). Retrieved September 24, 2018, from <http://docs.guzzlephp.org/en/stable/>
- [4] Otwell, T. (n.d.). Laravel - The PHP Framework For Web Artisans. Retrieved September 14, 2018, from <https://laravel.com/docs/5.4/>
- [5] jQuery Tutorial. (n.d.-e). Retrieved September 27, 2018, from <https://www.w3schools.com/jquery/>
- [6] The DomCrawler Component (Symfony Docs) [Forum post]. (n.d.-e). Retrieved October 5, 2018, from https://symfony.com/doc/current/components/dom_crawler.html
- [7] Correa, L. (2018). Desarrollo de una aplicación para búsqueda y extracción de texto desde la web para el Análisis de los Sentimientos (Tesis de pregrado). Universidad Católica De La Santísima Concepción, Concepción, Chile.

8. ANEXO: ESTIMACIÓN POR PUNTOS DE CASOS DE USO

8.1. CALCULAR UAW (UNADJUSTED ACTOR WEIGHTS)

Clasificación de actores

Actor	Tipo de actor	Peso
Usuario	Complejo	3
		UAW = 3

Tabla 19 Cálculo de pesos de actores sin ajuste

8.2. CALCULAR UUCW (UNADJUSTED USE CASE WEIGHTS)

Clasificación de casos de uso

Caso de uso	Tipo de caso de uso	Peso
Iniciar sesión	Simple	5
Registrar Usuario	Simple	5
Activar Cuenta	Simple	5
Crear Extractor	Promedio	10
Ejecutar Extractor	Simple	5
Programar Extractor	Promedio	10
Extraer desde Twitter	Simple	5
Descargar CSV	Simple	5
		UUCW = 50

Tabla 20 Cálculo de pesos de casos de uso sin ajuste

8.3. CALCULAR TCF (TECHNICAL COMPLEXITY FACTOR)

<i>Descripción</i>	Peso	Factor (Complejidad percibida)	Total
<i>Sistema distribuido.</i>	2	0	0
<i>Rendimiento</i>	1	0	0
<i>Eficiencia del usuario final.</i>	1	5	5
<i>Procesamiento interno complejo.</i>	1	5	5
<i>El código debe ser reutilizable.</i>	1	5	5
<i>Facilidad de instalación.</i>	0,5	5	2,5
<i>Facilidad de uso.</i>	0,5	5	2,5
<i>Portabilidad.</i>	2	0	0
<i>Facilidad de cambio.</i>	1	5	5
<i>Concurrencia.</i>	1	0	0
<i>Incluye objetivos especiales de seguridad.</i>	1	2	2
<i>Provee acceso directo a terceras partes.</i>	1	2	2
<i>Se requiere facilidades especiales de entrenamiento a usuario.</i>	1	3	3
			<i>TC = 32</i>

Tabla 21 Cálculo de factor de complejidad técnico

8.4. CALCULAR ECF (ENVIRONMENTAL COMPLEXITY FACTOR)

<i>Descripción</i>	Peso	Factor (complejidad percibida)	Total
<i>Familiaridad con el modelo de proyecto utilizado.</i>	1,5	1	1,5
<i>Experiencia en la aplicación.</i>	0,5	2	1
<i>Experiencia en orientación a objetos.</i>	1	5	5
<i>Capacidad del analista líder.</i>	0,5	3	1,5
<i>Motivación.</i>	1	3	3
<i>Estabilidad de los requerimientos</i>	2	5	10
<i>Personal part-time</i>	-1	0	0
<i>Dificultad del lenguaje de programación</i>	-1	2	-2
			EC =20

Tabla 22 Cálculo de factor de complejidad ambiental

8.5. CALCULAR UCP (USE CASE POINTS)

1. Calcular UUCP (unadjusted use case points)

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 50$$

$$UUCP = 53$$

2. Calcular TCF (technical complexity factor)

$$TCF = 0,6 + (0,01 * TComplexity)$$

$$TCF = 0,6 + (0,01 * 32)$$

$$TCF = 0,92$$

3. Calcular ECF (environmental complexity factor)

$$ECF = 1,4 + (-0,03 * EComplexity)$$

$$ECF = 1,4 + (-0,03 * 20)$$

$$ECF = 0,8$$

4. Finalmente calcular UCP (use case points)

$$UCP = UUCP * TCF * EF$$

$$UCP = 53 * 0,92 * 0,8$$

$$UCP = 39,008$$

8.6. ESTIMACIÓN FINAL

<i>UCP</i>	39,008
<i>LOE</i>	20
<i>Esfuerzo total = (UCP * LOE)</i>	780,16 H/H

Tabla 23 Estimación de esfuerzo total

9. ANEXO: RECURSOS UTILIZADOS API TECO

9.1. RECURSOS REQUERIDOS

Cuentas			
Método	Descripción	Ruta	Parámetros
POST	Registrar nuevo usuario	/api/signup	New User : Object
POST	Activar cuenta de usuario y envía correo de bienvenida	/api/activation	email : String , token : String , login : String
POST	Login con usuario activo o no activo (sin servicios)	/api/login	login : String , password : String
POST	Envía email con código para reiniciar contraseña	/api/forgot	email: String
POST	Cambiar contraseña por una nueva	/api/reset	Reset Password : Object
POST	Cambiar email por email nuevo, se envía código de activación	/api/invalid	Change Email : Object

Tabla 24 Servicio Cuentas

Usuarios			
Método	Descripción	Ruta	Parámetros
GET	Obtener información de usuario o información de otros usuarios	/api/user/{login}	login : String
GET	Obtener extractores de usuario o extractores compartidos	/api/user/{login}/extractors	login : String
POST	Añadir un extractor de usuario	/api/user/extractors/add	extractorData : Object
POST	Eliminar extractor de usuario	/api/user/extractors/delete	extractorID : String
POST	Modificar extractor de usuario	/api/user/extractors/update	extractorDataUpdate: Object

Tabla 25 Servicio Usuarios

Extractores			
Método	Descripción	Ruta	Parámetros
GET	Todos los extractores compartidos	/api/extractors	
GET	Todos los extractores compartidos por nombre	/api/extractors/name/:name	name : String
GET	Todos los extractores compartidos por URL	/api/extractors/url/{url}	url : String
GET	Un Extractor por ID	/api/extractors/id/{id}	id : String
POST	Ejecutar un extractor específico	/api/extractors/executeExtractor	extractorID : String
POST	Ejecutar un nodo específico desde un extractor	/api/extractors/executeNode	extractorID : String , nodeName : String
POST	Ejecutar Extractor Twitter con una consulta	/api/extractors/twitter	querySearch String
POST	Ejecutar una prueba de un extractor personalizado	/api/extractors/testExtractor	testExtractorData : Object

POST	Ejecutar una prueba a un nodo personalizado	/api/extractors/testNode	testNodeData: Object
------	---	--------------------------	-----------------------------

Tabla 26 Servicio extractores

Programar Extracciones			
Método	Descripción	Ruta	Parámetros
GET	Todas las extracciones personales programadas.	/api/schedule	
GET	Una programación personal por ID.	/api/schedule/{id}	id : String
POST	Agregar una nueva extracción programada para un Nodo específico	/api/schedule/addNode	extractorID String , nodeName String , atHour, atMinute
POST	Agregar una nueva extracción programada para un Extractor específico	/api/schedule/addExtractor	extractorID : String , atHour, atMinute
POST	Eliminar una extracción programada específica por ID	/api/schedule/delete	scheduleID : String

Tabla 27 Servicio Programar Extractores

9.2. OBJETOS REQUERIDOS

New User Object		
Parámetro	Tipo	Elemento interno
login :	String	Alphanumeric
email :	String	Valid Email
password :	String	Alphanumeric
passwordRepeat :	String	Alphanumeric

Tabla 28 Objeto Nuevo Usuario

Reset Password Object		
Parámetro	Tipo	Elemento interno
login :	String	Alphanumeric
email :	String	Valid Email
password :	String	Alphanumeric
passwordRepeat :	String	Alphanumeric
token :	String	

Tabla 29 Objeto Reiniciar Contraseña

Change Email Object		
Parámetro	Tipo	Elemento interno
login :	String	Alphanumeric
email :	String	Valid Email
password :	String	Alphanumeric
newEmail :	String	Valid Email

Tabla 30 Objeto Cambiar Email

Extractor Data Object		
Parámetro	Tipo	Elemento interno
name :	String	
url :	String	
description :	String	
nodes :	Array	Node Objects
shared :	Boolean	

Tabla 31 Objeto Datos Extractor

Extractor Data Update Object		
Parámetro	Tipo	Elemento interno
Id :	String	
name :	String	
url :	String	
description :	String	
nodes :	Array	Node Objects
shared :	Boolean	

Tabla 32 Objeto Actualizar Datos Extractor

Node Object		
Parámetro	Tipo	Elemento interno
name :	String	
container :	String	
description :	String	
elements :	Array	Element Objects

Tabla 33 Objeto Nodo

Element Object		
Parámetro	Tipo	Elemento interno
tag :	String	
selector :	String	

Tabla 34 Objeto Elemento

Test Extractor Data Object		
Parámetro	Tipo	Elemento interno
name :	String	
url :	String	
nodes :	Array	Node Objects

Tabla 35 Objeto Probar Datos Extractor

Test Node Data Object		
Parámetro	Tipo	Elemento interno
name :	String	
container	String	
elements :	Array	Element Objects

Tabla 36 Objeto Probar Nodo