



## IIC3103 - Taller de Integración

Departamento Ciencia de la Computación  
Escuela de Ingeniería  
Pontificia Universidad Católica

# Enunciado Tarea 1

## Objetivo

El objetivo de esta tarea es desplegar un sitio web que se conecte a una API Rest para obtener información y mostrarla al público.

## Trabajo a realizar

Se debe construir un sitio web, disponible de forma pública, que muestre información sobre los episodios y personajes de Rick and Morty. Como fuente de datos deberá utilizarse la API Rest de The Rick And Morty API.

### URL API

<https://rickandmortyapi.com/>

### URL DOCUMENTACIÓN

<https://rickandmortyapi.com/documentation>

## Navegación

El usuario debe ser capaz de ingresar al sitio y completar el siguiente flujo:

1. En la página principal, se deben listar todos los capítulos, con su nombre, fecha en que estuvo al aire, y código de episodio.
2. Al hacer click en el nombre de un episodio, se debe mostrar una nueva página con toda la información que entrega la API relacionada al episodio, incluyendo los nombres de todos sus personajes, y sin incluir id, url, ni fecha de creación.
3. Al hacer click en un personaje, deberá llevar a una nueva página que muestre toda la información de ese personaje, incluyendo los nombres de todos sus lugares y episodios, y mostrando la imagen del personaje. No debe incluir id, url, ni fecha de creación.
4. Al hacer click en un lugar, deberá llevar a una nueva página que muestre toda la información de ese lugar, incluyendo los nombres de todos sus personajes (residentes), y sin incluir id, url, ni fecha de creación.

- 
- Nota: Los episodios, personajes y lugares enlistados en las distintas vistas deberán mostrarse con el nombre de cada ítem, y no en la forma de url que entregan algunas llamadas de la API.

## Búsqueda

En todo el sitio deberá mostrarse una barra de búsqueda, que al ingresar un texto, busque sobre los nombres de los diferentes elementos de la API (episodios, personajes y lugares).

Al hacer un click sobre un resultado de búsqueda, el sitio debe mostrar toda la información adicional pedida en la sección de Navegación para cada página particular.

## Consideraciones

### Cuidado con el API Rate Limit

La API tiene un rate limit (tasa de requests) de 10.000 consultas por día. De ser sobrepasado este límite, recibirás un error “429” en todas tus consultas por un período de 12 horas.

### Ten en cuenta la paginación

La API automáticamente entrega las respuestas en páginas con 20 elementos cada una. Considera esto a la hora de hacer tus consultas iterando sobre el número de páginas de ser necesario.

## Otros requisitos

### Arquitectura de la solución

El sitio web podrá estar construido sobre un framework MVC tal como Rails (Ruby), Django (Python), Sails (Nodejs) u otro, o como un sitio estático que consuma directamente la API.

En el primer caso, se recomienda hacer un deploy en Heroku, sistema de servidores Cloud que tiene una capa gratuita con capacidades suficientes para el desarrollo de esta tarea.

En el segundo caso, se puede utilizar un servicio como Github Pages o similar para publicar el sitio.

### Uso de librerías

El objetivo de esta tarea es que logren hacer la implementación de consultas a la API. Por este motivo, no está permitido el uso de librerías que ya cuenten con esta implementación. Es decir, librerías que hagan las consultas directas a la API, porque ustedes deben hacerlas. Por lo tanto,

---

no se podrá utilizar ninguna librería de la sección Libraries de la documentación. Esta restricción incluye también librerías no oficiales y no publicadas en el sitio de la API pero que también se hayan construido especialmente para acceder a la información de esta API.

**Sí** está permitido el uso de librerías para hacer consultas HTTP. A diferencia de las librerías no permitidas, estas ayudan a construir la consulta pero ustedes deberán estructurarlas con los parámetros correspondientes. Algunos **ejemplos** de librerías que les podrían servir para esto:

- Python: Requests
- Javascript: Axios
- Ruby on Rails: HTTP Gem

## Versionamiento del código

El sitio y todo su código fuente deberá estar versionado en un repositorio en Github.

## Evaluación

### Entregables

Cada alumno deberá entregar, mediante un formulario publicado en el sitio del curso, las siguientes *url's*:

- URL de inicio del sitio implementado
- URL del repositorio Github

### Fecha de entrega

La tarea se deberá entregar el día 5 de abril antes de las 23:59 hrs.

### Requisitos mínimos

Las tareas que no cumplan las siguientes condiciones no serán corregidas y serán evaluados con la nota mínima:

- El sitio deberá ser público, accesible desde cualquier dispositivo conectado a internet.
- El código deberá estar versionado en su totalidad en un repositorio Git
- El sitio debe reflejar fielmente el código entregado en el repositorio
  - Se recomienda configurar un deploy automático en Heroku asociado al repositorio Github

---

## Penalizaciones

Se descontará 0,2 puntos de la nota de la tarea por cada hora de atraso en la entrega, contados a partir de la fecha estipulada en el punto anterior.

Cualquier intento de copia, plagio o acto deshonesto en el desarrollo de la tarea, será penalizado con nota 1,1 de acuerdo a la política de integridad académica del DCC.

---

## Anexo 1: Recursos adicionales

A continuación, se presentan una serie de links que le podrán ser útiles para el desarrollo de la tarea.

	Ruby on Rails	Python Django	Node
Cursos y documentación	Instalar: <a href="https://gorails.com/setup/osx/10.15-catalina">https://gorails.com/setup/osx/10.15-catalina</a>  Documentación: <a href="https://guides.rubyonrails.org/getting_started.html">https://guides.rubyonrails.org/getting_started.html</a>  Curso: <a href="https://www.codecademy.com/learn/learn-rails">https://www.codecademy.com/learn/learn-rails</a>	Documentación: <a href="https://docs.djangoproject.com/es/3.0/intro/tutorial01/">https://docs.djangoproject.com/es/3.0/intro/tutorial01/</a>	Tutorial: <a href="https://itnext.io/a-new-and-better-mvc-pattern-for-node-express-478a95b09155">https://itnext.io/a-new-and-better-mvc-pattern-for-node-express-478a95b09155</a>
Despliegue de una app en Heroku	<a href="https://devcenter.heroku.com/articles/getting-started-with-rails5#store-your-app-in-git">https://devcenter.heroku.com/articles/getting-started-with-rails5#store-your-app-in-git</a>	<a href="https://devcenter.heroku.com/articles/django-app-configuration">https://devcenter.heroku.com/articles/django-app-configuration</a>	<a href="https://devcenter.heroku.com/articles/deploying-nodejs">https://devcenter.heroku.com/articles/deploying-nodejs</a>

## Resolviendo dudas

### StackOverflow

Foro más popular para resolver dudas sobre código, *frameworks* y lenguajes de programación. Antes de preguntar, asegúrate que la pregunta no se ha realizado anteriormente.

<http://stackoverflow.com/>

### Resolución de dudas en Canvas

Se habilitará un foro en Canvas<sup>1</sup> para resoluciones de dudas específicas de esta tarea.

---

<sup>1</sup> Link sitio del curso: <https://cursos.canvas.uc.cl/courses/5563>

---

## Anexo 2: Pauta de corrección

Se evaluarán los siguientes ítemes, de acuerdo a las funcionalidades requeridas.

Funcionalidad	Descripción
Entrega	Se entrega url del sitio y esta responde correctamente.
	Se entrega url del repositorio y este refleja fielmente el sitio publicado.
	El código solo utiliza librerías permitidas. Cumple con lo especificado en la sección Uso de librerías.
	La información de la API se extrae haciendo las llamadas HTTP cada vez, sin uso de bases de datos pre-pobladas.
Home	Al ingresar al sitio, se muestran los 31 episodios de las primeras tres temporadas de la serie.
	Para cada capítulo, se muestra su nombre, fecha en que estuvo al aire, y código de episodio.
	Existe un link que da la opción de ver información detallada para cada episodio.
Episodio	Se muestra toda la información de un solo episodio, incluyendo su nombre, fecha en que estuvo al aire, código de episodio, y los nombres de todos los personajes que aparecen en ese episodio.
	Al hacer clic en el nombre de un personaje, lleva a la vista de detalle correspondiente al personaje.

---

<b>Personaje</b>	Al entrar en la vista de detalle de un personaje, se muestra toda la información de este, incluyendo su nombre, estado, especie, tipo, género, nombre del lugar de origen, nombre de su locación, imagen, y los nombres de todos los episodios en los que aparece.
	Al hacer clic en el nombre de un lugar o episodio, lleva a la vista de detalle correspondiente.
<b>Lugar</b>	Al entrar en la vista de detalle de un lugar, se muestra toda la información de este, incluyendo su nombre, tipo, dimensión, y nombres de todos los personajes residentes.
	Al hacer clic en el nombre de un personaje, lleva a la vista de detalle correspondiente al personaje.
<b>Búsqueda</b>	Al realizar una búsqueda, se muestran todos los resultados de la búsqueda para nombres de episodios, personajes, y lugares. Cada resultado permite ir a ver más información de este.