



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 consumir una API WEB para obtener información que esta entrega, y desplegar un sitio web que muestra al público la información que la API proporciona.

Trabajo a realizar

Se debe construir un sitio web, que esté disponible de forma pública, que muestre la información que entregan los distintos servicios que componen la API.

URL API

<https://us-central1-taller-integracion-310700.cloudfunctions.net/tarea-1-2021-2/{canvasId}>

URL DOCUMENTACIÓN

<https://us-central1-taller-integracion-310700.cloudfunctions.net/tarea-1-2021-2/>

Para el uso de la API, cada estudiante deberá proporcionar su Id de Canvas (parámetro canvasId de la URL de la API), el cual se puede obtener ingresando a la siguiente URL desde el navegador, habiendo iniciado sesión previamente en Canvas:

<https://cursos.canvas.uc.cl/api/v1/users/self>

Parte 1 (25%): Consultar API

Cada alumno deberá consultar la API para responder preguntas respecto a los datos que los servicios publican. Las preguntas se publicarán el día antes de la entrega de la tarea.

Parte 2 (75%): Sitio web y navegación

Cada alumno deberá construir un sitio web, que permita visualizar los datos de la API. En particular, se pide que el usuario pueda completar los flujos que se describen a continuación.

Visualización de usuarios

1. Al ingresar, se deberá mostrar el listado de todos los usuarios que entrega la API. Los resultados de los usuarios vienen paginados en grupos de 10 resultados.
2. Para cada usuario, se deberá desplegar la opción de mostrar toda la información asociada al usuario.
3. Al hacer click en la opción de mostrar toda la información de un usuario, se deberá mostrar toda la información de tarjetas de crédito y direcciones que el usuario ha registrado.
4. Las ciudades deberán ser “clickeables”. Al hacer click sobre el nombre de una ciudad, se deberá ir a la página de detalle de la ciudad seleccionada (ver sección visualización de ciudades, punto 3).

Visualización de ciudades

1. Al ingresar, se deberá mostrar un listado de ciudades que entrega la API, donde se visualice el nombre de la ciudad y el país donde se ubica. Los resultados de las ciudades vienen agrupados en grupos de 10 resultados.
2. Para cada ciudad, se deberá desplegar la opción de mostrar toda la información asociada a la ciudad.
3. Al hacer click sobre la opción de mostrar toda la información de la ciudad, se deberá mostrar, además del nombre y país, los nombres de los usuarios que tienen direcciones registradas en la ciudad seleccionada.
4. Los usuarios deben ser “clickeables”. Esto quiere decir que cuando se hace click sobre el nombre de un usuario, se deberá dirigir a la vista de detalle del usuario (ver sección de visualización de usuarios, punto 4).

Búsqueda

En todo el sitio deberá mostrarse una barra de búsqueda, que al ingresar un texto, muestre los resultados de las búsquedas de:

1. Usuarios con nombre y/o apellido que correspondan a la búsqueda. Al seleccionar un resultado de usuario, se deberá dirigir a la vista de detalle del usuario (ver sección de visualización de usuarios, punto 5).
2. Nombres de ciudades que correspondan con la búsqueda. Al seleccionar un resultado de ciudad, se deberá ir a la página de detalle de la ciudad seleccionada (ver sección visualización de ciudades, punto 3).

Consideraciones

Ten en cuenta la paginación

Los servicios entregan respuestas en páginas de máximo 10 elementos. Considera esto a la hora de hacer tus consultas iterando sobre el número de páginas de ser necesario.

Los datos son dinámicos

Considera que los datos pueden cambiar: se pueden modificar o agregar nuevos datos, por lo que tu implementación deberá siempre mostrar los resultados que entrega la API.

Además, los resultados pueden diferir para los distintos estudiantes. Para la corrección, se considerarán los sets de datos correspondientes a cada alumno.

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. Para esta tarea no es necesario tener un backend o código de servidor de ningún tipo, por lo que una arquitectura de este tipo permite cumplir todo lo solicitado.

Versionamiento del código

El sitio y todo su código fuente deberá estar versionado en un repositorio en Github, el cual será provisto a través de Github Classrooms.

Para crear el repositorio, cada alumno deberá unirse al Classroom del curso, en la siguiente URL:

Github Classroom

<https://classroom.github.com/classrooms/37184944-iic3103-taller-de-integracion-2021-2>

Tarea 1

<https://classroom.github.com/a/voEEQxNZ>

Bases de datos

Para esta tarea, no es necesario el uso de base de datos, y no se recomienda su uso debido a que los datos de la API irán cambiando en el tiempo. Se espera que el sitio generado muestre siempre los resultados más actualizados posibles.

Evaluación

La tarea deberá realizarse en forma individual. La evaluación de la tarea consistirá en:

1. Parte 1: Completitud y exactitud de las respuestas entregadas a las preguntas que se realicen.
2. Parte 2: Metodología de evaluación entre pares.

Entregables

Cada alumno deberá entregar, mediante un formulario publicado en el sitio del curso, lo siguiente:

- URL de inicio del sitio implementado
- URL del repositorio Github
- Respuestas a las preguntas relacionadas a la parte 1.

Fecha de entrega

La tarea se deberá entregar el día 10 de septiembre a las 18:00.

Requisitos mínimos

Las tareas cuya parte 2 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 del sitio deberá estar versionado en su totalidad en el repositorio Git que será provisto
- El código en el repositorio debe reflejar fielmente el sitio entregado

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.

Documentación y ejemplos para lenguajes y frameworks específicos

	Ruby on Rails	Python Django	Node
Cursos y documentación	<p>Instalar: https://gorails.com/setup/osx/11.0-big-sur</p> <p>Documentación: https://guides.rubyonrails.org/getting_started.html</p> <p>Curso: https://www.codecademy.com/learn/learn-rails</p>	<p>Documentación: https://docs.djangoproject.com/es/3.1/intro/tutorial01/</p>	<p>Tutorial: https://itnext.io/a-new-and-better-mvc-pattern-for-node-express-478a95b09155</p>
Despliegue de una app en Heroku	https://devcenter.heroku.com/articles/getting-started-with-rails5#store-your-app-in-git	https://devcenter.heroku.com/articles/django-app-configuration	https://devcenter.heroku.com/articles/deploying-nodejs

Documentación general y otros ejemplos

Despliegue en Heroku	Documentación general sobre el despliegue de una aplicación en Heroku, para múltiples lenguajes y frameworks.	https://devcenter.heroku.com/categories/language-support
Despliegue en Github Pages	Documentación general sobre el despliegue en Github Pages.	https://pages.github.com/ https://www.codecademy.com/articles/f1-u3-github-pages
Consumir API con Javascript	Ejemplos para consumir una API desde un sitio con Javascript, utilizando XMLHttpRequest, fetch o Axios	https://medium.com/swlh/consuming-apis-using-javascript-c980abd36d59
Desplegar datos de una API	Desplegar datos de una API en un sitio web utilizando Javascript	https://w3collective.com/fetch-display-api-data-javascript/ https://www.digitalocean.com/community/tutorials/how-to-use-the-javascript-fetch-api-to-get-data https://www.javascripttutorial.net/javascript-fetch-api/
Sobre HTML	Cursos rápidos para construir un sitio web con HTML	https://www.freecodecamp.org/news/learn-html-in-5-minutes-ccd378d2ab72/ https://html.com/

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/>