



**Certified Tech
Developer**
The Ultimate Degree

Programación Imperativa

Segundo Parcial

Objetivo

Integrar los conocimientos adquiridos hasta el momento en la cursada.

Practicar la lectura, comprensión e interpretación de consignas y producir código que resuelva los enunciados propuestos.

Buenas prácticas

A la hora de la resolución no sólo es importante que el código funcione correctamente, sino también que sea legible y respete las buenas formas.

- Que las variables, métodos y funciones tengan nombres descriptivos
- Que se utilice camelCase donde corresponda
- Que se utilicen los métodos más adecuados para resolver cada problema
- Que el código produzca el resultado esperado a partir de los datos suministrados

Introducción

Una empresa que reúne Prestadores de Servicios de Salud (PSS) necesita crear un pequeño sistema que le permita facilitar la gestión de estos.

La empresa tiene una base de información que podremos encontrar en un array de objetos literales que representan las propiedades de cada profesional al comienzo del archivo.



Para comenzar

Deberán descargar la siguiente [PLANTILLA](#) donde encontrarán la estructura inicial que necesitan para una entrega exitosa.

- **plantilla-app.js**, será donde escribiremos nuestro código para resolver las consignas.

Consignas

- A. Crear un objeto literal que represente la empresa. Este objeto literal lo llamaremos por ejemplo **appProfesionales** . Agregar una propiedad llamada **profesionales** que contenga al array de profesionales.
- B. Agregar un método **listarProfesionales** que permita confeccionar un listado que facilite al usuario, leer los datos de los profesionales que estén dentro del array que se recibe por parámetro, de manera que imprima por consola los siguientes datos con este formato:
 - id 3 inactivo, Neumonología, Leanne Burch, consultas: 1, email: leanneburch@speedbolt.com
 - id 8 ok, Neumonología, Guerra Bright, consultas: 8, email: guerrabright@speedbolt.com
- C. Agregar un método **filtrarHabilitadosPorEspecialidad** que permita filtrar profesionales que estén habilitados, y además, por determinada especialidad recibida por parámetro.
 - Este método deberá retornar un array de profesionales habilitados por especialidad
 - Sólo recibe por parámetro la especialidad.
- D. Agregar un método **buscarPorID** que permita buscar dentro de la propiedad de nuestro objeto profesionales un profesional por la propiedad id, la misma deberá ser igual a un id que se envía como argumento al momento de invocarlo.
 - Este método deberá retornar un profesional.



- E. Agregar un método **realizarConsulta** que permita incrementar en 1 la cantidad de consultas de un profesional, el mismo deberá realizar los siguiente pasos.
- Recibe un parámetro, el id del profesional.
 - Reutiliza el método **buscarPorID** para obtener el profesional.
 - Una vez encontrado el profesional deberá cambiar el valor de la propiedad *cantidadConsultas* , incrementando su valor en 1
 - Al finalizar deberá retornar al profesional en cuestión
- F. Agregar un método **corregirEmails** que corrija los emails de todos los profesionales dado que algunos por error tienen en lugar del "@" el símbolo "#"